



Universidad  
Carlos III de Madrid

DEPARTAMENTO DE INFORMÁTICA

PROYECTO FIN DE CARRERA

Ingeniería Técnica Informática de Gestión

# ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN MODELO DE REFERENCIA DE PROCESOS PARA MEJORAR LA VALIDACIÓN Y VERIFICACIÓN DE PRODUCTOS SOFTWARE

AUTOR: Jessica Castellanos Fraga

TUTORES: Ana Sanz Esteban  
Javier Saldaña Ramos

Leganés, 20 de Junio de 2011

ESTA PÁGINA HA SIDO DEJADA EN BLANCO INTENCIONADAMENTE

Título: Análisis, Diseño e Implementación de un Modelo de Referencia de Procesos para mejorar la validación y verificación de productos software.

Autor: Jessica Castellanos Fraga

Director: Ana Sanz Esteban y Javier Saldaña Ramos

## EL TRIBUNAL

Presidente: Fuensanta Medina Domínguez

Vocal: Diana Marcela Vasquez Bravo

Secretario: José Luis López Cuadrado

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 20 de Junio de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

ESTA PÁGINA HA SIDO DEJADA EN BLANCO INTENCIONADAMENTE

# AGRADECIMIENTOS

*Gracias a mis niños, que desde el cielo, me han enviado todas las fuerzas que he necesitado durante la carrera y en el día a día y las cuales actualmente sigo recibiendo. Os quiero y os echo mucho de menos.*

*Gracias a mis padres, por brindarme la oportunidad de estudiar una carrera, apoyarme en mis estudios y animarme dándome cariño en mis malos momentos pero sobre todo por concederme una educación que me ha formado como persona y me ha inculcado unos maravillosos valores.*

*Gracias a Ana y a Javier, tutores del proyecto, por la confianza depositada en mí para la realización del presente proyecto. Por su ayuda y todo el tiempo que me han dedicado, por todo lo que he aprendido en los últimos meses y especialmente por el apoyo recibido.*

*Gracias a mi hermano Iván, que siempre será mi pequeñín, y desde que nació se convirtió en la alegría de mi casa y de mi vida.*

*Gracias a Carlos, por su gran apoyo , comprensión y ayuda incondicional durante toda la carrera y sobre todo por estos seis años tan bonitos, porque tú me transmites ese amor, tranquilidad y protección que necesito.*

*Gracias a mis abuelos, qué a pesar de la distancia, me dan todo su cariño, su admiración y sus palabras de aliento cuando las necesito.*

*Gracias a mi amigo y compañero Alberto, por tenderme una mano cuando la he necesitado. Estoy muy contenta de haber compartido contigo esta etapa.*

*Gracias a mis amigos y compañeros de universidad por haber disfrutado con ellos estos años de carrera en los que hemos sufrido tanto malos como buenos momentos.*

# RESUMEN

## **Palabras clave:**

Calidad, Aseguramiento de Calidad, Verificación y Validación, Pruebas software, Procesos, Modelo de Referencia de Procesos

## **Resumen:**

En la actualidad los productos finales continúan sin alcanzar el nivel de calidad requerido por el cliente. Uno de los principales problemas es la falta de disciplina en el desarrollo del proceso de las pruebas de software. Éstas suelen retrasarse hasta el final del proyecto, dónde el tiempo es reducido y, por tanto, no se realizan con la precisión que deberían.

Llevar a cabo un proceso de pruebas software eficaz es una tarea de alto coste que no todas las organizaciones pueden implantar. Es por ello que están surgiendo nuevas organizaciones dedicadas a la provisión de servicios de pruebas software. El objetivo de este PFC es ofrecer un modelo de referencia de procesos de verificación y validación que permita a las organizaciones establecer modelos de provisión de servicios de pruebas software. Este modelo será útil tanto para las organizaciones dedicadas a la provisión de pruebas software como para las organizaciones que se dediquen a mejorar la calidad de sus productos.

# ABSTRACT

## **Keywords:**

Quality, Software Quality Assessment, Verification & Validation, Software Testing, Reference Process Model

## **Summary:**

At present the final products still do not reach the level of quality required by the customer. One of the main problems is the lack of discipline in the development process of software testing. These are often delayed until the end of the project, where time is limited and therefore not carried out with precision as they should.

Conducting an effective software testing process is a costly task that not all organizations can deploy. That is why new organizations are emerging the provision of software testing services. The aim of the thesis is to provide a reference model of verification and validation processes to enable organizations to establish models for provision of software testing services. This model will be useful for organizations dedicated to providing software testing and for organizations dedicated to improving the quality of their products.

# ÍNDICE GENERAL

1.	INTRODUCCIÓN.....	14
1.1	Introducción .....	14
1.2	Objetivos .....	15
1.3	Fases de desarrollo.....	16
1.4	Medios empleados.....	16
1.5	Estructura de la memoria .....	17
1.6	Acrónimos .....	18
2.	ESTADO DEL ARTE .....	20
2.1	Primera aproximación: Calidad.....	20
2.2	Aseguramiento de la Calidad .....	24
2.3	Proceso de Aseguramiento de la Calidad .....	27
2.4	Verificación y Validación .....	31
2.5	Pruebas de Software .....	37
2.6	Modelos de Referencia de Procesos.....	41
2.6.1	Primera aproximación: Procesos .....	41
2.6.2	Modelo de Referencia.....	42
2.7	Externalización de las Pruebas Software .....	46
2.7.1	Outsourcing.....	46
2.7.2	Externalizar pruebas software .....	48
2.8	Conclusiones .....	50
3.	PROPUESTA DE MODELO DE REFERENCIA DE PROCESOS .....	52
3.1	Proceso de definición de las fases de desarrollo.....	52
3.2	Especificación de los procesos .....	53
3.3	Modelo de Referencia de Procesos .....	54
4.	IMPLEMENTACIÓN DEL MODELO .....	92
5.	CONCLUSIONES Y LÍNEAS FUTURAS.....	99
5.1	Conclusiones .....	99
5.2	Líneas Futuras .....	100



6.	PLANIFICACIÓN Y PRESUPUESTO .....	102
6.1	Planificación.....	102
6.2	Presupuesto .....	104
7.	GLOSARIO.....	106
8.	REFERENCIAS.....	109
	ANEXO 1: CHECKLIST PARA VERIFICAR CÓDIGO FUENTE .....	115
	ANEXO 2: CHECKLIST PARA LAS ACTIVIDADES DEL PRODUCTO DE TRABAJO .....	121
	ANEXO 3: CHECKLIST PARA LOS PRODUCTO DE TRABAJO.....	128
	ANEXO 4: ESPECIFICACIÓN DEL PROCESO ASEGURAMIENTO DE LA CALIDAD.....	136
	ANEXO 5: PLAN DE ASEGURAMIENTO DE LA CALIDAD.....	146
	ANEXO 6: PLAN DE GESTIÓN DE CONFIGURACIÓN .....	150
	ANEXO 7: PLAN DE PETICIÓN DE CAMBIO .....	156
	ANEXO 8: PLANTILLA PARA RECOGER LA INFORMACIÓN PARA LA AUDITORIA.....	160
	ANEXO 9: PLANTILLA PARA EL INFORME DE LA AUDITORIA.....	162
	ANEXO 10: INFORME PARA LA VERIFICACIÓN DEL CÓDIGO FUENTE .....	165
	ANEXO 11: INFORME DE INSPECCIÓN .....	168
	ANEXO 12: INFORME DE VERIFICACIÓN .....	172
	ANEXO 13: INFORME DE RESULTADOS .....	177

# ÍNDICE DE FIGURAS

Figura 1: Coste total de calidad.....	23
Figura 2: Actividades del grupo de SQA.....	26
Figura 3: Actividades principales de calidad .....	28
Figura 4: Gráfica Coste-Calidad.....	29
Figura 5: Niveles de Madurez del modelo CMMI .....	44
Figura 6: Comparación de las representaciones del Modelo CMMI.....	44
Figura 7: Modelo TMM .....	45
Figura 8: Estructura TMMi .....	46
Figura 9: Especificación de los procesos .....	53
Figura 10: Modelo de Referencia.....	55
Figura 11: Proceso Gestión de las Actividades de Pruebas.....	57
Figura 12: Actividades del proceso Gestión de las Actividades de Pruebas .....	57
Figura 13: Actividad1: Establecer los objetivos y políticas .....	58
Figura 14: Actividad 2: Planificación de las pruebas.....	59
Figura 15: Actividad 3: Especificación de casos de prueba.....	60
Figura 16: Actividad 4: Preparar el entorno de pruebas.....	60
Figura 17: Actividad 5: Ejecutar los casos de prueba.....	61
Figura 18: Actividad 6: Evaluar las pruebas realizadas .....	62
Figura 19: Proceso Pruebas de Aceptación.....	62
Figura 20: Actividades del proceso Pruebas de Aceptación .....	63
Figura 21: Actividad 1: Planificación de las pruebas.....	64
Figura 22: Actividad 2: Especificación de los casos de prueba .....	65
Figura 23: Actividad 3: Preparar el entorno de pruebas de aceptación .....	65
Figura 24: Actividad 4: Ejecutar los casos de prueba.....	66
Figura 25: Actividad 5: Evaluar pruebas de aceptación .....	67
Figura 26: Proceso Pruebas de Sistema .....	67
Figura 27: Actividades del proceso de Pruebas de Sistema.....	68
Figura 28: Actividad 1: Planificación de las pruebas.....	69
Figura 29: Actividad 2: Especificación de casos de prueba.....	70
Figura 30: Actividad 3: Preparar el entorno de de pruebas del sistema .....	70
Figura 31: Actividad 4: Ejecutar los casos de prueba.....	71
Figura 32: Actividad 5: Evaluar pruebas de sistema .....	72
Figura 33: Proceso Verificación Código Fuente .....	72
Figura 34: Actividades del proceso Verificación Código Fuente .....	73
Figura 35: Actividad 1: Seleccionar los proyectos a verificar .....	73
Figura 36: Actividad 2: Planificar el proceso de verificación .....	74
Figura 37: Actividad 3: Realizar verificación formal.....	75
Figura 38: Actividad 4: Evaluar características de calidad .....	76
Figura 39: Proceso Verificación de Productos de Trabajo .....	76
Figura 40: Actividades del proceso Verificación de Productos de Trabajo .....	77
Figura 41: Actividad 1: Seleccionar los proyectos a verificar .....	77

Figura 42: Actividad 2: Planificar el proceso de revisión .....	78
Figura 43: Actividad 3: Realizar Inspección.....	79
Figura 44: Actividad 4: Realizar Walkthroughs .....	80
Figura 45: Actividad 5: Verificar procedimientos de gestión.....	80
Figura 46: Actividad 6: Elaborar informe de resultados .....	81
Figura 47: Proceso Gestión de la Configuración .....	81
Figura 48: Actividades del proceso Gestión de la Configuración.....	82
Figura 49: Actividad 1: Identificación de la Configuración .....	83
Figura 50: Actividad 2: Control de Cambios.....	83
Figura 51:Actividad 3:Contabilidad de Estado .....	84
Figura 52: Actividad 4: Auditoría de la Configuración .....	85
Figura 53: Proceso Aseguramiento de la Calidad .....	85
Figura 54: Actividades del proceso Aseguramiento de la Calidad .....	86
Figura 55: Actividad 1: Las actividades de calidad a realizar .....	87
Figura 56: Actividad 2: Planificar las actividades de calidad.....	87
Figura 57: Actividad 3: Realizar las actividades de calidad .....	88
Figura 58: Actividad 4: Documentar y resolver las incidencias.....	89
Figura 59: Actividad 5: Documentar y resolver las lecciones aprendidas.....	90
Figura 60:EPG.Visión General: Proceso Verificación de Productos de Trabajo .....	95
Figura 61:EPG.Actividades del Proceso Verificación de Productos de Trabajo .....	96
Figura 62:EPG.Criterios de Salida del proceso Verificación de Productos de Trabajo.....	96
Figura 63:EPG.Actividad1 del proceso Verificación de los Productos de Trabajo .....	97
Figura 64:EPG.Tareas de la Actividad 1 del proceso Verificación de los Productos de Trabajo .....	97
<b>Figura 65: Planificación inicial del proyecto .....</b>	<b>102</b>
<b>Figura 66: Planificación real del proyecto .....</b>	<b>103</b>
Figura 67: Presupuesto .....	104
Figura 68: Producto de Trabajo: Plan de Aseguramiento de la Calidad .....	148
Figura 69: Producto de Trabajo: Plan de Gestión de Configuración.....	154
Figura 70: Producto de Trabajo: Plan de Gestión de Cambio .....	158
Figura 71: Producto de Trabajo: Plantilla para recoger información de la auditoría .....	160
Figura 72: Producto de trabajo: Plantilla para el informe de la auditoría .....	163
Figura 73: Producto de Trabajo: Informe para verificar el código fuente .....	166
Figura 74: Producto de Trabajo: Informe de inspección.....	170
Figura 75: Producto de Trabajo: Informe de verificación .....	175
Figura 76: Producto de Trabajo: Informe de resultados.....	178

# ÍNDICE DE TABLAS

Tabla 1: Categoría de costes .....	24
Tabla 2: Técnicas Informales .....	33
Tabla 3: Técnicas Estáticas .....	34
Tabla 4: Técnicas Dinámicas .....	35
Tabla 5: Técnicas Formales .....	36
Tabla 6: Checklist verificar código fuente .....	119
Tabla 7: Checklist documento de requisitos .....	121
Tabla 8: Checklist análisis estructurado .....	123
Tabla 9: Checklist análisis orientado a objetos .....	124
Tabla 10: Checklist diseño estructurado .....	125
Tabla 11: Checklist diseño orientado a objetos .....	126
Tabla 12: Checklist documento de requisitos .....	129
Tabla 13: Checklist Análisis del Sistema .....	130
Tabla 14: Checklist Diseño del Sistema .....	131
Tabla 15: Checklist Plan de Proyecto .....	132
Tabla 16: Checklist Plan de Pruebas del Proyecto .....	134

# CAPÍTULO 1

## INTRODUCCIÓN

## **1. INTRODUCCIÓN**

En este primer capítulo se incluye una breve explicación sobre calidad y la necesidad del Aseguramiento de la Calidad en las organizaciones software. Además, se trata la necesidad de realizar los procesos de verificación y validación sobre los productos mediante la externalización de las pruebas software con el fin de elaborar un Modelo de Referencia que sirva de guía en cuanto a que actividades de verificación y validación se deben realizar para mejorar la calidad de un producto software.

### **1.1 Introducción**

La mejora de la calidad en productos o servicios es una actividad cada vez más demandada en las organizaciones. Actualmente existen organizaciones donde sus proyectos fracasan o no terminan con el éxito que deberían, por lo que cada vez se preocupan más de seguir un plan por el equipo de trabajo durante el desarrollo de los productos que garantice la calidad del mismo.

Los productos o servicios ofrecidos por una organización deben cumplir las necesidades y expectativas del cliente además de establecer una mejora continua sobre ellos. Para ello, el equipo de trabajo debe certificar que un producto reúne las características necesarias para satisfacer todos los requerimientos establecidos y por tanto garantizar que se trata de un producto adecuado y fiable en calidad.

En base a lo anterior, podemos establecer una serie de funcionalidades básicas que oferta el Aseguramiento de la Calidad:

- Reducir, eliminar y lo más importante, prevenir las deficiencias de calidad de los productos a obtener.
- Alcanzar una razonable confianza en que las prestaciones y servicios esperados por el cliente o el usuario queden satisfechas.

El Aseguramiento de la Calidad tiene una fuerte relación con la verificación y validación de productos software debido a que para que éstos se realicen con la calidad esperada es necesario incorporar actividades de calidad a lo largo de sus procesos de desarrollo, que permita tener un mayor control de los errores y una reducción de costes. Para conseguir tales aspectos se emplean las pruebas software, ya que nos permiten identificar posibles fallos de implementación, calidad o usabilidad.

Por tanto, se puede decir que las Pruebas de Software constituyen un mecanismo o herramienta que permite inyectar rápidamente calidad en los productos.

Sin embargo, diseñar, definir e implantar un proceso de pruebas software en la organización, es una tarea muy compleja y de elevado coste. Se necesita disponer de personal capacitado para el desarrollo de las tareas, la infraestructura adecuada para la realización de los mismos, y el proceso correctamente definido e implementado.

La inversión necesaria para alcanzar este propósito es tan elevada que muchas organizaciones no pueden abordarla. Como consecuencia de esta situación, han surgido empresas dedicadas a la provisión de servicios de pruebas software. De este modo, aquellas organizaciones que no dispongan del presupuesto necesario para abordar la implantación de un proceso de pruebas, podrá subcontratar estos servicios y disfrutar de los beneficios que se obtienen del mismo.

Así, en este proyecto fin de carrera se propone un modelo de referencia de procesos de verificación y validación que permitirá a las organizaciones establecer modelos de provisión de servicios de pruebas software. Este modelo podrá ser implantado tanto en organizaciones cuya actividad de negocio sea la provisión de servicios relacionados con la verificación y validación, cómo en organizaciones de desarrollo de software que deseen mejorar la calidad de sus productos y/o su proceso de pruebas.

## **1.2 Objetivos**

El objetivo principal de este proyecto fin de carrera es el desarrollo de un Modelo de Referencia de procesos de Verificación y Validación de productos software que especifique el conjunto de buenas prácticas a seguir con el propósito de mejorar la calidad de los productos software.

Para alcanzar este objetivo principal se han definido los siguientes objetivos específicos:

1. Analizar los conceptos teóricos asociados con calidad y aseguramiento de la calidad, qué requerimientos deben cumplir y cómo alcanzarlos.
2. Analizar el concepto de proceso, saber qué es y porque es necesario.
3. Analizar los conceptos de Validación y Verificación, en qué consiste y cuáles son sus objetivos.
4. Identificar qué procesos son los necesarios para mejorar y garantizar la calidad en los productos.
5. Diseñar el modelo de referencia propuesto.
6. Definir los procesos que componen el modelo.
7. Implementar el Modelo de Referencia de Procesos.

### 1.3 Fases de desarrollo

El desarrollo de este proyecto se ha llevado a cabo en diferentes fases:

- **Identificar documentación relevante:** En esta fase se ha identificado toda la información necesaria para poder desarrollar con claridad todos los conceptos tratados en este proyecto.
- **Analizar documentación:** En esta fase se estudió de manera detallada toda la información identificada para conocer cuál de toda esa información era útil para llevar a cabo el desarrollo del proyecto.
- **Diseñar los procesos:** A partir de la información analizada, se diseñaron los procesos de verificación y validación que iban a asegurar la calidad en los productos eliminando los posibles defectos.
- **Definir los procesos:** Una vez diseñado los procesos, se pasó a definir en qué iba a consistir cada uno de los procesos para que fueran capaces de garantizar al cliente productos de alta calidad que cumplen sus expectativas.
- **Seguimiento de los procesos:** En esta fase se lleva a cabo un seguimiento continuo de los procesos diseñado así como de todos los conceptos tratados en este proyecto, con el fin de poder realizar correcciones durante la realización del proyecto.
- **Evaluar el proyecto:** En esta fase se realiza una evaluación de todo el proyecto una vez este ha sido finalizado y en la cual se observa si el proyecto cumple con sus objetivos.

### 1.4 Medios empleados

En la elaboración de este proyecto ha sido empleados diversos medios. Para obtener la información deseada se ha hecho uso de diferentes libros que se encuentran en la biblioteca de la universidad así como diferentes páginas web. (*Véase apartado REFERENCIAS*). Además ha sido empleado un ordenador que contiene el paquete Office con Procesador de textos Microsoft Word 2007 para elaborar toda la documentación, Microsoft SourceSafe, que es una herramienta de control de versiones para elaborar la guía electrónica, Microsoft InfoPath, que es una herramienta para desarrollar formularios de datos basados en XML y ha sido utilizado para insertar los datos en la guía electrónica de procesos. Además de una impresora, cartuchos de tinta para la impresora, un escáner, folios y material de oficina.



## **1.5 Estructura de la memoria**

En este apartado se describe cómo está estructurado el presente documento, explicando brevemente el contenido de cada capítulo:

- **Capítulo 1: Introducción**
  - Este capítulo contiene una breve reseña para situar al lector en el contexto en el que se enmarca este proyecto así como los objetivos definidos.
- **Capítulo 2: Estado del Arte**
  - En este capítulo se proporciona un enfoque global sobre las diferentes áreas a las que hace referencia este proyecto: Calidad y SQA, Verificación y Validación, Procesos; que permita comprender el modelo propuesto y la relevancia del mismo por las Organizaciones Software.
- **Capítulo 3: Propuesta de Modelo de Referencia de Procesos de Verificación y Validación**
  - En este capítulo se especifica el modelo propuesto, a través de la descripción de las actividades y tareas que definen cada uno de los procesos identificados.
- **Capítulo 4: Implementación del Modelo**
  - En este capítulo se describe como se ha llevado a cabo la implementación del modelo propuesto, haciendo uso de una plataforma tecnológica que permite el soporte de Guías Electrónicas de Procesos.
- **Capítulo 5: Conclusiones y Líneas Futuras**
  - En este capítulo, se exponen las conclusiones más importantes obtenidas tras la realización del proyecto. Estas se han obtenido tras un proceso de reflexión acerca de los aspectos que han caracterizado el desarrollo del mismo. Además, se plantea un conjunto de líneas de trabajo a partir de las cuales se podría ampliar el alcance de la solución desarrollada en el marco de este proyecto fin de carrera.
- **Capítulo 6: Planificación y Presupuesto**
  - En este capítulo, se presenta el presupuesto total del proyecto indicando tanto los costes materiales como el coste del personal además de la planificación del proyecto organizada por fases.
- **Capítulo 7: Glosario**
  - En este capítulo, se presenta alfabéticamente cada una de las palabras o términos utilizados en la realización del proyecto.
- **Capítulo 8: Referencias**
  - Este capítulo incluye las referencias bibliográficas utilizadas a lo largo del desarrollo de este trabajo.

- **ANEXOS:**
  - Se incluye una sección de anexos que recoja información adicional que complemente la solución desarrollada.

## **1.6 Acrónimos**

<b>Acrónimo</b>	<b>Definición</b>
<b>QA</b>	Aseguramiento de la Calidad
<b>EC</b>	Elemento de Configuración
<b>EPG</b>	Guía Electrónica de Procesos
<b>GC</b>	Gestión de Configuración
<b>PGC</b>	Plan de Gestión de Configuración
<b>PQA</b>	Plan de Aseguramiento de Calidad

## CAPÍTULO 2

### ESTADO DEL ARTE

## 2. ESTADO DEL ARTE

### 2.1 Primera aproximación: Calidad

Actualmente, la satisfacción hacia el uso de un producto puede marcar una gran diferencia en el mercado de productos similares. Es así como el desarrollo de productos que satisfacen las expectativas de los clientes y usuarios harán la diferencia entre dos organizaciones que desarrollan productos que compiten en el mercado. La preocupación por ofrecer productos acompañados de altos niveles de calidad no es una actividad nueva.

El desarrollo de productos software no está ausente de ofrecer calidad. Dicho nivel de calidad, incluido en los productos, considera muchas actividades dentro del desarrollo de los proyectos software. La gestión de la calidad dentro de este tipo de proyectos puede estandarizarse dentro de la organización y certificarse a la comunidad de clientes. Antes de empezar hablar acerca de en que podría consistir la calidad de los productos software, se debería definir qué es lo que se entiende por calidad.

Si analizamos el concepto de calidad en nuestros días, es un término que encontramos en multitud de contextos como por ejemplo, calidad en la alimentación, calidad en los servicios y en la producción. Se habla de calidad en todos los productos que van a parar al cliente y cuyo fin es buscar una sensación positiva, transmitiendo la idea de que algo es mejor.

El concepto de calidad representa más bien una forma de hacer las cosas en las que fundamentalmente, predomina la preocupación por satisfacer al cliente y por mejorar, día a día, procesos y resultados. Dicho concepto ha evolucionado hasta convertirse en una forma de gestión que introduce el concepto de mejora continua en cualquier organización y a todos los niveles de la misma, y que afecta a todas las personas y a todos los procesos.

Pero, ¿Cómo definimos formalmente calidad?: Se puede definir según la norma ISO 9000 como *“Grado en el que un conjunto de características inherentes cumple con los requisitos”* [1]. De esta forma se podría decir que la calidad de los productos puede medirse como una comparación de sus propiedades y características.

Retrocediendo en el tiempo, es a partir de la II Guerra Mundial, cuando las naciones en guerra necesitaron mejorar sus procedimientos de fabricación y, sobre todo, la calidad de sus productos para atender la demanda industrial. Esto provocó un aumento en las técnicas comunes como la inspección de los productos y la elaboración de especificaciones técnicas.

Hacia la mitad del primer lustro de los años 40 la producción en masa aumentó y por tanto se hacía inviable la inspección de los productos al 100 por 100. En esa época W. Shewhart difundió en los Estados Unidos la aplicación de los métodos estadísticos al campo de la Calidad, extendiéndose el Control Estadístico de la Calidad a todas las grandes empresas. No obstante se observó que los productos resultaban defectuosos debido a que las especificaciones no estaban relacionadas ni ajustadas al producto. Se puede decir que

una consecuencia de la II Guerra Mundial fue la aparición del concepto de Calidad como “Conformidad a unas Especificaciones”. [2]

En 1943 **Ishikawa** define el concepto de calidad como: “*Calidad significa calidad del producto, calidad de trabajo, calidad del servicio, calidad de información, calidad de proceso, calidad de la gente, calidad del sistema, calidad de la compañía, calidad de objetivos, etc.*” [6]. Desarrolla el Diagrama Causa-Efecto como herramienta para el estudio de las causas de los problemas. Parte de que los problemas no tienen causas únicas, sino que suelen ser, según su experiencia, un cúmulo de causas. Sólo hay que buscar esta multiplicidad de causas, colocarlas en su diagrama (también conocido como de “espina de pescado”, ya que su forma nos la recuerda) formando familias de causas a las que aplicar medidas preventivas selectivas. [3] Los Círculos de la Calidad (también entendidos como de Control de la Calidad), están en la base del pensamiento de Ishikawa como una de las herramientas para el mejoramiento continuo y la puesta en práctica de la Calidad Total [5].

En 1950 **Deming** visitó Japón donde dio una serie de conferencias sobre Control de Calidad. Deming define calidad como: “Grado predecible de uniformidad y fiabilidad a bajo coste y adecuado a las necesidades del mercado” [3]. Mejoró el círculo de calidad propuesto por Shewhart, el cual consiste en localizar el problema y atacarlo de raíz, a través de 4 etapas las cuales son “Planear, Hacer, Verificar y Actuar”. El Círculo de Calidad se transforma en un proceso de mejora continua, ya que se analiza cada parte del proceso para ver cuál es la problemática y esto nos ayuda a conocerlo mejor y evitar futuros errores [4].

Posteriormente **Philip Crosby** comenzó su trabajo como profesional de la calidad en 1952 en una escuela médica. Su carrera la inició en una planta de fabricación en línea donde decidió que su meta sería enseñar que ser buenos en prevenir problemas es más provechoso que ser bueno en solucionarlos. Crosby define calidad como: “*Cumplimiento de requerimientos de usuarios*”. Los requerimientos tienen que estar claramente establecidos para que no haya malentendidos. Su lema es “*Hacerlo bien a la primera vez y conseguir cero defectos*” [6]. En 1960 Crosby es conocido por ser el creador de conceptos como Zero Defects (cero defectos) o Buck-a-Day (aprovecha el día). Parte de la idea de que hacer mal las cosas cuesta dinero mientras que hacerlas bien no agrega nada al coste del producto. [7]

Después el cliente comienza a exigir más calidad. Entonces se comienza a buscar que el grado de adaptación de un producto a su diseño sea el óptimo. En esta época se hace el Control de Calidad, en el sentido de inspección de las características de un producto y de satisfacer las necesidades técnicas y de producción. De este modo, la calidad se identifica con la ausencia de defectos ofreciendo productos con excelente calidad. Pero resulta que aunque el producto cumpla las especificaciones del diseño, no es aceptado por el mercado. Entonces surge la necesidad de cambiar el sistema de gestión y surge la Gestión de la Calidad. [7]

El concepto de calidad se mide mediante el grado de satisfacción de las necesidades del cliente. Los objetivos, por lo tanto, serán satisfacer al cliente, mantener la calidad, reducción

de los costes y mejorar la competitividad de la empresa. En este momento surge el Aseguramiento de la Calidad. El concepto básico de este Sistema de Calidad supone garantizar el nivel de calidad del producto, esto es, que el resultado de la actividad de la empresa sea el que se pretende y no una sorpresa. El punto débil de este sistema es que no contempla la mejora del producto, ni define sistemas para captar la voz del cliente.

Para mejorar este punto débil surge la mejora continua, herramienta utilizada de diferentes maneras en cada empresa según sus necesidades y métodos de trabajo. La mejora continua está basada en una serie de pequeñas mejoras que van haciendo avanzar poco a poco a la empresa en los diferentes aspectos.

Las empresas más comprometidas en materia de calidad han comenzado recientemente a incorporar un sistema de gestión denominado Gestión de Calidad Total. Este proceso supone integrar el concepto de calidad en todas las fases del proceso y a todos los departamentos que tienen alguna influencia en la calidad final del proceso y/o servicio prestado al cliente.

Actualmente, los “gurús” de la calidad llegan aun más lejos. Taguchi define la calidad como el grado de pérdida para la sociedad. El objetivo, por lo tanto es buscar el método de producción que supone un coste mínimo para la sociedad. En este concepto entran otro tipo de consideraciones, como pueden ser las relaciones con el medio ambiente y la satisfacción de los trabajadores.

Esto hace suponer que en un futuro el concepto de calidad se identifique con la satisfacción por el trabajo bien hecho. Los objetivos buscados pasarían a ser la satisfacción interna (empresa), la satisfacción externa (cliente y sociedad en general), y una alta competitividad en un mercado en el que la calidad se considerará como un derecho. [7]

Sin embargo, conseguir que el concepto de la calidad se identifique con un trabajo bien hecho conlleva una serie de costes debidos a que existe una alta relación entre costes, calidad, inversiones y mejoramiento, especialmente en el mejoramiento de la calidad. Dichos costes se clasifican de la siguiente manera:

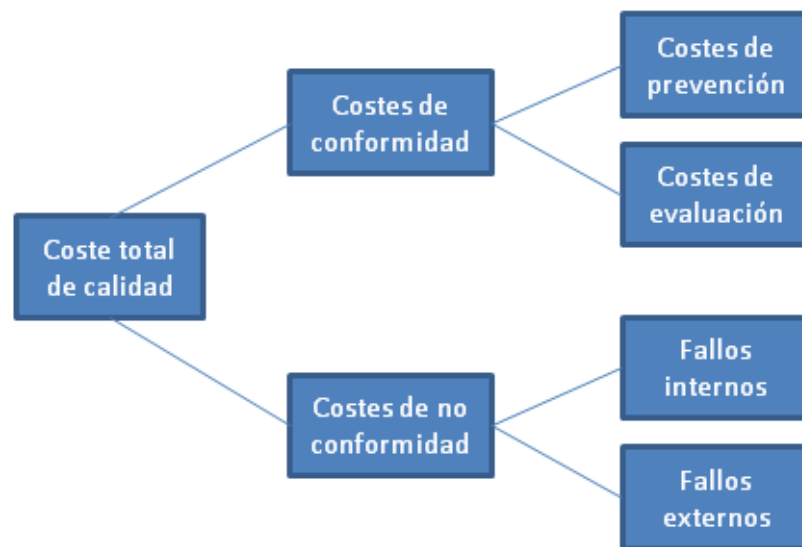


Figura 1: Coste total de calidad

A continuación, en base al esquema anteriormente planteado se explica en qué consiste cada uno de estos costes.

COSTE		DESCRIPCIÓN
Costes de no conformidad	<b>Costes de fallos internos</b>	Son los asociados con defectos (errores, no conformidad, etc.) que se encuentran antes de transferir el producto al cliente. Pueden citarse: Desperdicios, Retrabajo, Análisis de fallos, Materiales de desperdicio y retrabajo, Inspección del 100%, Re-inspección y volver a probar, Pérdidas de proceso evitables y Rebajas.
	<b>Costes de fallos externos</b>	Son los relacionados con defectos que se encuentran después de enviar el producto al cliente. Estos costes desaparecerían si no hubiera defectos. Pueden ser: Costes de garantía, Conciliación de quejas, Material regresado y Concesiones.

<b>Costes de conformidad</b>	<b>Costes de evaluación</b>	Son aquéllos en los que se incurre al determinar el grado de conformidad con los requerimientos de calidad, como por ejemplo: Inspección y prueba al recibir, Inspección y prueba en proceso, Inspección y prueba final, Auditorías de la calidad del producto, Mantenimiento del equipo de prueba, Inspección y prueba de materiales y servicios y Evaluación del inventario.
	<b>Costes de prevención</b>	Son los generados al mantener los costes de fallos y de apreciación al mínimo. Algunos son: Planeación de la calidad, Revisión de nuevos productos, Control de procesos, Auditorías de calidad, Evaluación de la calidad del proveedor y Entrenamiento.

**Tabla 1: Categoría de costes**

Para conseguir que estos costes que se acarrean en la obtención de calidad en un producto sean mínimos o nulos, es necesario llevar a cabo un proceso de Aseguramiento de la Calidad (QA).

## **2.2 Aseguramiento de la Calidad**

Se puede decir que el Aseguramiento de la Calidad son todas las actividades planificadas y sistemáticas que se necesitan para proporcionar la confianza adecuada de que un producto tiene las propiedades necesarias para satisfacer necesidades implícitas o explícitas. [8]

El objetivo principal que persigue es dar al cliente productos con calidad. Para ello, se deberá realizar un conjunto de actividades como reducir, eliminar y, lo más importante, prevenir las deficiencias de calidad en los productos a obtener o alcanzar una razonable confianza en que las prestaciones y servicios esperados por el cliente o el usuario queden satisfechas. [9]

Asegurar implica evaluar un proceso o actividad, identificar las oportunidades de mejora, planear y diseñar cambios, introducir cambios, reevaluar la actividad o proceso, documentar cambios y verificar que la actividad o proceso se realiza de acuerdo a la documentación formal existente.



Las normas base del Aseguramiento de Calidad son las ISO 9000 y en relaciones contractuales las ISO 9001, 9002, 9003 y 9004. Estas normas buscan en general que todo lo que se haga esté documentado y escrito, lo que esté documentado se aplique, lo que se aplique sea efectivo, los productos malos no lleguen nunca al cliente, los problemas no se repitan y se prevengan, se controlen y apliquen los cambios (contratos, diseño, compras, fabricación, documentos, instalación y servicios).

Si hablamos de los beneficios más relevantes obtenidos al aplicar Aseguramiento de Calidad percibimos que son muchos y variados. Algunos de ellos son los siguientes [10]:

- **Se detectan problemas rápidamente.** Es posible identificar problemas en tempranas etapas del desarrollo de productos de software, ayudando al desarrollador a corregirlos inmediatamente y poder avanzar con más rapidez.
- **Se crean y se siguen estándares de trabajo.** Con apoyo del proceso de aseguramiento de calidad, se pueden establecer estándares tan diversos como son los de codificación o de documentación, los cuales apoyan a uniformizar y consolidar el proceso de desarrollo.
- **Se verifica que los objetivos individuales vayan acordes con los objetivos de la organización.** Se busca y se recomienda centrarse en los objetivos planificados y expectativas del cliente de manera que los requerimientos expuestos por usuarios finales estén alineados con los objetivos globales de la empresa, facilitando así el logro de los mismos y la integración total de los usuarios a la organización.
- **Se recomiendan métodos para realizar el trabajo.** Las prácticas de aseguramiento de calidad, como son muy robustas ya que aplican técnicas muy completas de medición, pueden proponer en un momento dado qué métodos se ajustan más a la naturaleza del producto a ser desarrollado, teniendo como efecto final que el producto tenga más posibilidades de ser un producto con calidad.
- **Se evita incurrir en costes innecesarios.** Como un efecto generalizado de algunos de los puntos mencionados con anterioridad, la práctica de procesos de aseguramiento de calidad lleva a las organizaciones a evitar costes no deseados como pueden ser todos aquellos ocasionados por mantenimiento correctivo.
- **Se planifica la calidad.** Está claro que el concepto de calidad no es algo que se da de una manera automática e impredeciblemente. Es algo que se busca. Por lo mismo, se debe de planificar, construir e implantar en el producto además de asegurarse que se logra y se mantiene.

Un instrumento al servicio del Aseguramiento de Calidad que permite desarrollar las actividades de forma disciplinada y ordenada es el Plan de SQA.

Este es un documento de apoyo a las actividades de calidad que pretende identificar que actividades de calidad que se van a realizar, los estándares a aplicar, los productos a revisar, los procedimientos a seguir en la obtención de los distintos productos y la normativa

para informar de los defectos detectados a sus responsables además de realizar el seguimiento de los mismos hasta su corrección. [9]

Habitualmente, las actividades relevantes al Aseguramiento de la Calidad son realizadas por el grupo de Aseguramiento de Calidad, que tiene la responsabilidad de planificar, supervisar, guardar registros, analizar y reportar la garantía de calidad además de auxiliar al equipo de desarrollo a obtener un producto final de alta calidad.

El grupo de SQA se encargará también de efectuar algunas de las siguientes actividades [11]:

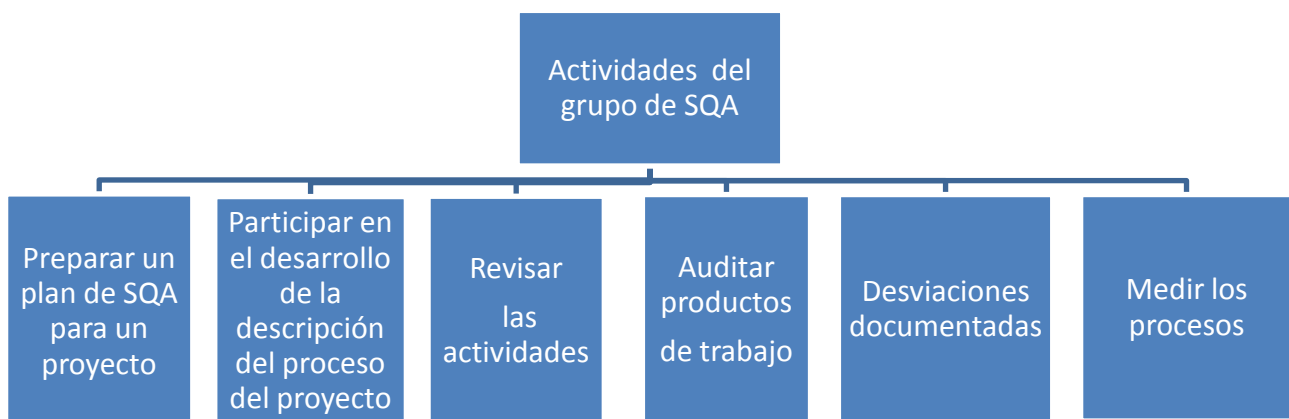


Figura 2: Actividades del grupo de SQA

A continuación se ofrece una breve explicación de cada una de las actividades:

**Preparar un plan de SQA para un proyecto:** Todos los proyectos de desarrollo de software deben preparar un Plan SQA del proceso de prueba, el cual debe guiar las actividades realizadas por el grupo SQA, así como a los ingenieros de software implicados en el proyecto.

**Participar en el desarrollo de la descripción del proceso del proyecto:** En la definición del proyecto deben quedar establecidos los estándares, técnicas y metodologías que serán aplicadas en el mismo, en el flujo de trabajo de prueba, quedando reflejadas en el punto de técnicas y metodologías del Plan de SQA.

**Revisar las actividades para verificar que se ajustan al proceso definido:** Se debe llevar a cabo mediante una reunión previamente planificada donde quede reflejada las técnicas y metodologías a utilizar en el Plan SQA del proceso de prueba además de describir las tareas y responsabilidades del plan SQA. Los artefactos de prueba deben revisarse en pequeñas porciones para encontrar el mayor número de errores y lograr un buen desarrollo del mismo.

**Auditar productos de trabajo seleccionados para verificar que se ajustan con los definidos como parte del proceso:** Para las auditorías se utilizará un acta de revisión centrado en los objetivos de cada actividad, además de en el Registro de Auditorías.

**Garantiza que las desviaciones en los productos de trabajo están documentadas:** Las desviaciones son detectadas quedando reflejadas en una lista de desviaciones y además será informado el jefe del grupo de SQA. Estas desviaciones pueden ser: desviaciones en los atributos de calidad o desviaciones en el cronograma del proyecto. Las desviaciones deben revisarse periódicamente hasta que se solucionen y la documentación de las mismas debe ser controlada y administrada por un integrante del grupo de SQA.

**Medir los procesos.** Se incluyen métricas para evaluar los procesos del flujo de trabajo de prueba.

Con el propósito de garantizar que las actividades de calidad desarrolladas son las adecuadas, y que se están llevando a la práctica, es necesaria la definición e implantación de un proceso de aseguramiento de la calidad en la organización.

### **2.3 Proceso de Aseguramiento de la Calidad**

El proceso de Aseguramiento de la Calidad sirve para certificar que un producto reúne las características necesarias para satisfacer todos los requerimientos establecidos y por tanto garantizar que se trata de un producto adecuado y fiable en calidad.

Su objetivo principal es llevar a cabo el seguimiento de las actividades de calidad comprometidas internamente en el proyecto. Para ello, será necesario verificar que las actividades realizadas son desarrolladas correctamente y hacen más eficiente el aprovechamiento de los recursos.

Para realizar este proceso, se debe pasar por una serie de actividades encargadas de verificar que el producto cumple con la calidad y con los requisitos estimados. Estas actividades se basan en determinar y planificar los productos que van a ser revisados, para posteriormente documentar las posibles incidencias que serán resueltas y anotadas para que en futuros productos no vuelvan a cometerse. Se lleva a cabo en las diferentes fases por las que pasa el desarrollo de un producto, comenzando por la fase de **obtención de las necesidades del cliente**, debido a que un producto con calidad es el que satisface las necesidades del cliente. Para que esto sea hay que tener en cuenta las siguientes actividades que son representadas en el siguiente organigrama:

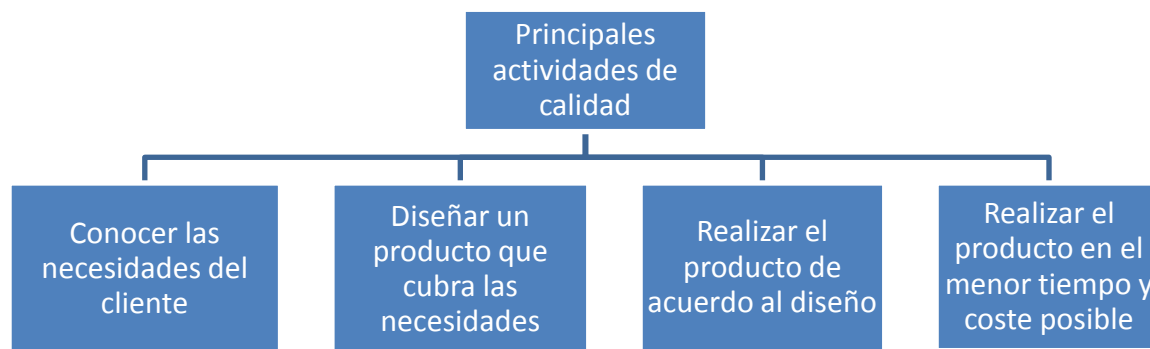


Figura 3: Actividades principales de calidad

Como se ha mencionado anteriormente, el producto debe ser acorde al diseño y corresponder además de con las especificaciones planificadas, con las exigencias del proyecto. Por tanto, otra de las fases por las que debe de pasar el Aseguramiento de la Calidad es la fase del **diseño**. El diseño de un nuevo producto se puede resumir en estas etapas:

- **Elaborar el proyecto:** Su calidad dependerá de la viabilidad de fabricar y producir el producto según las especificaciones planificadas.
- **Controlar el proceso de diseño:** El proceso de diseño debe ser controlado para asegurarnos que los resultados son los previstos.
- **Definir la técnica del producto:** Dicha definición se lleva a cabo a través de la técnica AMFE. Esta técnica es usada para definir, identificar y eliminar problemas antes de que afecten al cliente.

Finalmente, se debe llevar a cabo la **Evaluación de la Calidad** de un producto y obtener como resultado si dicho producto satisface las especificaciones y exigencias del proyecto. Para ello, se contará con los siguientes indicadores:

- **La calidad de conformidad:** Es la medida en que un producto se corresponde con las especificaciones diseñadas, y concuerda con las exigencias del proyecto.
- **La calidad de funcionamiento:** Indica los resultados obtenidos después de utilizar los productos fabricados.

Además de seguir estos indicadores, es imprescindible llevar a cabo un **Control de Calidad** sobre el producto debido a que durante el desarrollo se pueden realizar continuas observaciones acerca del cumplimiento de las tareas.

El objetivo principal de control de calidad es ofrecer calidad objetiva en cuanto a la forma en cómo se está desarrollando un producto, es decir, una vigilancia permanente a todo el proceso de desarrollo y ciclo de vida del producto. Esta meta puede alcanzarse

mediante frecuentes inspecciones a las metodologías de trabajo y al uso de herramientas o mediante las revisiones de prototipos y testeo exhaustivo de los productos finales.

Además, llevar a cabo un control de la calidad permite realizar las rectificaciones pertinentes al desarrollo en cuanto este empieza a desviarse de sus objetivos. Estas rectificaciones son posibles gracias a una revisión de las etapas superiores, las cuales nos permiten obtener un aprendizaje, al observar las salidas de cada etapa hasta llegar al producto final.

La revisión, así como cada etapa realizada, debe generar documentación sobre el diseño de los procesos de la etapa así como de los resultados obtenidos en cada etapa (y que servirá de entrada a la etapa siguiente). Esto permite mejorar los procesos cuya calidad es deficiente al mismo tiempo que ofrece calidad máxima en aquellos procesos que la organización ha llevado a cabo en el desarrollo del producto.

En el control de calidad se debe tener presente los costes totales que esta involucra. Si reflexionamos sobre que tareas se deben realizar en este control, puede observarse que es necesario llevar a cabo tareas de búsqueda de problemas, testeo, realimentación, rectificación, elaboración, modificación y estudio de la documentación; entre otras actividades. Todas ellas tienen costes involucrados (incluso puede darse la inclusión de equipos destinados al aseguramiento de la calidad: los grupos SQA). Pero debe existir un compromiso, ya que un excesivo coste en el control de la calidad puede hacer que este proceso se torne ineficiente. Por otra parte, la mejora de la calidad implica reducir los costes, ya que se tendría un cierto nivel de calidad asegurado.

Finalmente, y como consecuencia de la naturaleza del proceso de desarrollo de productos, asegurar la calidad en las primeras etapas del proceso de desarrollo involucra que los costes del control en las etapas posteriores tenderá a disminuir al tener menos aspectos que controlar. A continuación, se observa la relación existente entre coste-calidad:

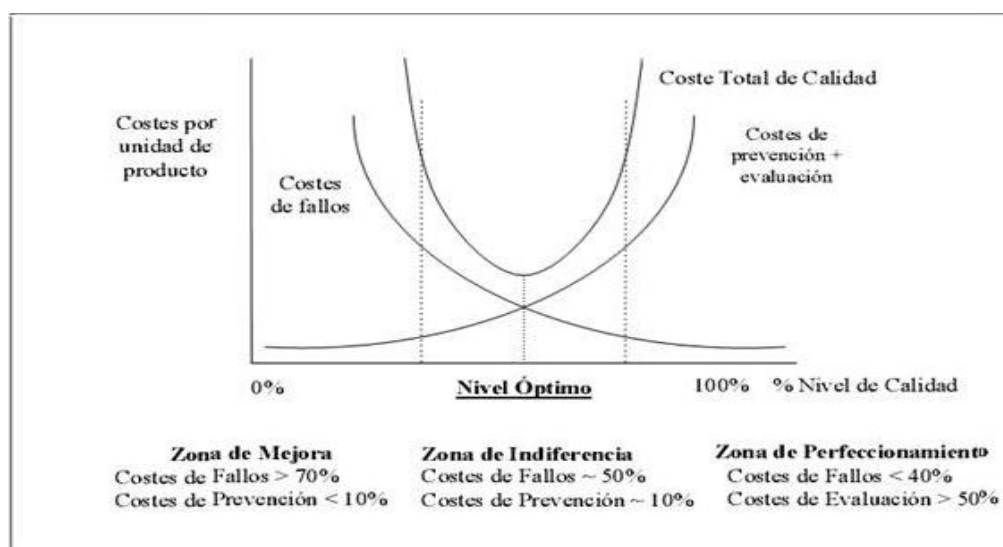


Figura 4: Gráfica Coste-Calidad

Las tres curvas representantes de los costes de calidad, de no calidad y del coste total de la calidad están funcionalmente relacionadas con la calidad de conformidad. En el gráfico de la figura 4, se observa que los costes por fallos decrecen de modo continuo con el aumento de la calidad de conformidad dibujando un recorrido que va desde cero cuando el 100% de los productos cumplen las especificaciones hasta infinito cuando el 100% de los productos son defectuosos. Los costes de prevención y de evaluación son cero cuando el 100% de unidades son defectuosas y se elevan asintóticamente conforme se alcanzan niveles próximos a los cero defectos. La curva del coste total de la calidad tiene un mínimo que se alcanzará cuando los esfuerzos adicionales de prevención y evaluación no se justifican ya que ayudan a minimizar los costes de los fallos producidos. [12]

La curva del coste de la calidad total se ha dividido en tres zonas:

- **Primera zona: Zona de mejora de la calidad.** Se caracteriza por un sistema orientado hacia la corrección de errores. Como consecuencia, la estructura del coste total de la calidad está dominada por los costes de fallos, que constituyen más del 70% del total, mientras que los costes de prevención no llegan al 10%. Una empresa situada en esta zona puede reducir su coste total de la calidad realizando proyectos de mejora con la utilización racional de recursos logrando así el valor óptimo de costos de calidad.
- **Segunda zona: Zona de indiferencia.** Están las empresas con sistemas de calidad basados en el control.  
Como consecuencia de esta política los costes de los fallos alcanzan, en general, alrededor del 50% de los costes de la calidad, mientras que los costes de prevención oscilan alrededor del 10%.
- **Tercera zona: Zona de perfeccionamiento.** La política de la empresa está dirigida a la evaluación, en busca de aproximar la calidad de conformidad al 100%. Esta búsqueda de la perfección provoca una contracción importante de los costes por fallos hasta por debajo del 40% del coste total de la calidad, a costa del aumento de los costes de evaluación, que pueden superar el 50%. [12]

En definitiva, la calidad de los productos viene determinada por los procesos que se encargan de generarlo. Es decir, el proceso es el que va a determinar la calidad de un producto desde que comienza su desarrollo hasta el final. Para los usuarios finales la calidad es una característica propia del producto y por ello no se debe descuidar los procesos.

Todo lo anteriormente tratado es ampliado a la **calidad de productos software** de manera que la calidad permite definir acciones planeadas y sistemáticas que requieren ser seguidas para lograr conformidades finales en los procesos de desarrollo, ahorro de costes y satisfacción de los clientes.

Por último, cabe mencionar que el uso de estándares facilita y mejora el control de calidad. Se puede citar los siguientes estándares: ISO 12207 e ISO 9000.

**ISO/IEC 12207** Information Technology / Software Life Cycle Processes, es el estándar para los procesos de ciclo de vida del software de la organización ISO. Establece un proceso de ciclo de vida que incluye procesos y actividades que se aplican desde la definición de requisitos, pasando por la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso [8].

**ISO/IEC 9000** designa un conjunto de normas sobre calidad y gestión continua de calidad, establecidas por ISO. Se pueden aplicar en cualquier tipo de Organización o actividad orientada a la producción de bienes o servicios. Las normas recogen tanto el contenido mínimo como las guías y herramientas específicas de implantación, como los métodos de auditoría. ISO 9000 detalla la manera en que una organización opera, cuáles son sus estándares de calidad, sus tiempos de entrega y niveles de servicio [8].

## **2.4 Verificación y Validación**

Como se comentaba en la introducción, una forma de incrementar la calidad de los productos es mediante las pruebas de software. Las actividades de verificación y validación forman parte del proceso de SQA. Dado que el modelo de referencia que se propone en este trabajo versa sobre los procesos de verificación y validación, se considera oportuno profundizar en ellos. Para ello se debe saber que se entiende por Verificación y Validación.

- **Validación:** El propósito de la validación es demostrar que un producto o componente cumple con su uso intencionado cuando es puesto en su entorno. Es decir, asegura que “has construido el sistema correcto”. [14]
- **Verificación:** La verificación se refiere a la construcción correcta del sistema. Se puede definir verificación como el proceso de determinar si la lógica operacional del modelo (programa de ordenador) se corresponde con la lógica del diseño. En términos más simples, consiste en determinar si hay errores en el programa. [14]

El objetivo fundamental de la verificación es intentar determinar si los productos de una actividad se ajustan a los requisitos o a las condiciones impuestas en actividades anteriores, mientras que la validación se encarga de controlar si el producto cumple con los requisitos establecidos además de aumentar a un nivel aceptable la credibilidad del producto para que sea aceptado por los usuarios finales.

Para llevar a cabo la validación se debe realiza las siguientes actividades:

**1) Preparación para la validación.**

- Seleccionar los productos y componentes que van a ser validados, así como los métodos de validación que se van a utilizar para cada uno.
- Establecer y mantener el entorno necesario para soportar la validación.
- Establecer y mantener los procedimientos y criterios de validación.

**2) Validar los productos o componentes.**

- Validar los productos o componentes que hayan sido seleccionados anteriormente.
- Analizar los resultados obtenidos de la validación.

Mientras que para realizar una buena verificación, las actividades a realizar son las siguientes:

**1) Preparación para la verificación.**

- Seleccionar los productos de trabajo que se van a verificar y los métodos de verificación que van a ser utilizados para cada uno.
- Establecer y mantener el entorno necesario para soportar la verificación.
- Establecer y mantener los procedimientos y criterios de verificación para los productos de trabajo que hayan sido seleccionados.

**2) Desarrollo de revisiones por pares.**

- Preparación de las revisiones por pares para los productos seleccionados.
- Realizar las revisiones por pares en los productos seleccionados e identificar los resultados obtenidos.
- Analizar la preparación de los datos de las revisiones, la realización de las mismas y los resultados obtenidos en cada una de ellas.

**3) Verificar los productos de trabajo seleccionados.**

- Desarrollar la verificación de los productos seleccionados.
- Analizar los resultados obtenidos de todas las actividades de verificación.

Tanto Verificación como Validación son procesos que ayudan a una organización a cumplir con sus plazos, costes y calidad sobre el producto; además de mejorar la transición de la fase de desarrollo a la de puesta en producción. De esta manera cuando los productos llegan a esta fase final ofrecen una mejor respuesta a las necesidades y satisfacción de los usuarios finales proporcionando confianza a los responsables del desarrollo del nuevo producto. Se ahorra tiempo y el cliente está más satisfecho. [16]

En cuanto a las principales técnicas utilizadas en el proceso de Verificación y Validación, podemos establecer la siguiente clasificación: Informales, Estáticas, Dinámicas y Formales. [17]



Las **técnicas informales** son las más utilizadas habitualmente. Se basan en el razonamiento humano sin ningún tipo de formalismo matemático, lo que no implica la falta de estructura o líneas de guía formalizadas para el desarrollo de la técnica. Entre ellas se pueden considerar:

<b><u>Técnicas Informales</u></b>	
<b>Tipos</b>	<b>Descripción</b>
Comprobaciones de la documentación	Comprobar que toda la documentación de los productos generada es correcta.
Pruebas de Turing	Se basa en que un Juez humano entable una conversación con un ser humano y una máquina (a la vez) e intente establecer cuál es la máquina. Se considera que una máquina podrá superar la Prueba de Turing cuando sea capaz de comportarse tal como un humano y el juez no pueda discernir entre sus interlocutores.
Walkthroughs	El walkthrough debe intentar identificar defectos y considerar posibles soluciones en cualquier producto software.

**Tabla 2: Técnicas Informales**

En las **técnicas estáticas** la valoración se hace en base a las características del modelo estático y del código fuente. No requieren ejecución en una máquina. Existen herramientas automáticas para ayudar al proceso. Entre ellas se pueden considerar:

<b><u>Técnicas Estáticas</u></b>	
<b>Tipos</b>	<b>Descripción</b>
Grafos causa-efecto	El uso de grafos de causa-efecto es una técnica de casos de prueba que proporciona una concisa representación de las condiciones lógicas y sus correspondientes acciones.
Revisiones estáticas de código	Comprobar la conformidad de un código sin ejecutarlo.
Inspecciones	Son un método para verificar y validar un producto manualmente.
Revisiones	Pretenden detectar manualmente defectos en cualquier producto del desarrollo.

**Tabla 3: Técnicas Estáticas**

Las **técnicas dinámicas** necesitan la ejecución del modelo para poder evaluarlo basándose en los resultados de la ejecución. La mayoría de ellos requieren instrumentación del modelo. Algunas de ellas son:

<b><u>Técnicas Dinámicas</u></b>	
<b>Tipos</b>	<b>Descripción</b>
Pruebas de aceptación	<p>Son las únicas pruebas que son realizadas por los usuarios. Podemos distinguir entre dos pruebas:</p> <p><b>Pruebas alfa:</b> Las realiza el usuario en presencia de personal de desarrollo del proyecto haciendo uso de una máquina preparada para tal fin.</p> <p><b>Pruebas beta:</b> Las realiza el usuario después de que el equipo de desarrollo les entregue una versión casi definitiva del producto.</p>
Comprobación de aserciones	Esta técnica consiste en transmitir que el código siempre debe de cumplirse, de forma que si algo va mal se detecte cuanto antes.
Pruebas de conformidad <ul style="list-style-type: none"> <li>❖ Autorización</li> <li>❖ Prestaciones</li> <li>❖ Seguridad</li> <li>❖ De cumplimiento de estándares</li> </ul>	Tienen como objetivo verificar una hipótesis sobre el valor de un determinado parámetro.
Pruebas de interfaz <ul style="list-style-type: none"> <li>❖ De datos</li> <li>❖ De interfaz del modelo</li> <li>❖ De paso de mensajes</li> </ul>	Consiste en probar la interfaz correspondiente para garantizar que cumple los estándares y requerimientos definidos.

**Tabla 4: Técnicas Dinámicas**

Por último, las **técnicas formales** son aquellas que se encargan de descubrir errores en la función, lógica o implementación de cualquier producto mediante una reunión de revisión donde el equipo de trabajo evalúa el producto y señala las mejoras necesarias. Las técnicas formales más importantes son: [18]

<b><u>Técnicas Formales</u></b>	
<b>Tipos</b>	<b>Descripción</b>
Auditorias	Son las actividades para determinar el cumplimiento de los requisitos, planes y contrato. Este proceso puede ser empleados por dos partes cualesquiera, donde una parte (la auditora) audita los productos software o actividades de otra parte (la auditada)
Inspecciones	Es un tipo muy formal de revisión donde los revisores están siguiendo un proceso bien definido para encontrar defectos

**Tabla 5: Técnicas Formales**

Para concluir este apartado, cabe destacar que aunque de manera conceptual los conceptos de Verificación y Validación sean distintos, existe una fuerte relación entre ellos debido a que es un proceso de control que asegura que un producto cumple con su especificación y satisface las necesidades del usuario. Por tanto una correcta estrategia de Verificación y Validación ayudará a mejorar la calidad de los productos y reducirá los costes de corrección de errores.

## 2.5 Pruebas de Software

Las pruebas son un proceso relevante dentro del ciclo de vida de desarrollo software. A través de estas pruebas se consigue la detección de fallos del producto, un producto sin fallos será un producto de mayor calidad; siendo su coste económico y tiempo, los más elevados dentro del desarrollo. Una mejora que ofrece el proceso de pruebas es obtener un producto de mayor o igual calidad, en un menor tiempo y con un decremento de su coste; por lo que las Organizaciones comienzan a tratar de mejorar la calidad del proceso. En torno a este tema, han aparecido diferentes modelos de mejora que tratan de guiar a los usuarios proporcionando los pasos o actividades a realizar para mejorar y controlar dicho proceso.

Las pruebas software existen desde el nacimiento del software, ya que todo sistema necesita ser probado antes de ser utilizado. Sin embargo, el concepto de pruebas software ha ido evolucionando a través del tiempo.

Hasta 1957, no existía ninguna diferencia entre pruebas y depuración del software. D.Gelperin y B.Hetzel [19] definen esta etapa como *“The Debugging-Oriented Period”*. Los programadores escribían sus programas y comprobaban su funcionamiento, tratando a los conceptos de comprobar, depurar y probar del mismo modo, ya que no se encontraban claramente definidos. En 1949, Alan Turing [20] escribe el que es considerado como primer artículo sobre pruebas software. En él trata la pregunta: ¿Cómo podríamos conocer si un programa manifiesta inteligencia? *“How would we know that a program exhibits intelligence?”*, en otras palabras, si los requisitos son para construir un programa entonces la pregunta sería: ¿Cómo podríamos conocer si un programa satisface sus requisitos? – *“How would we know that a program satisfies its requirements?”*. Turing dió una definición de un test operacional para el comportamiento inteligente de un programa de ordenador. Este test requiere del comportamiento del programa y de un sistema de referencia (un humano) que no se pueda distinguir de un tester.

En 1957, Charles Baker avanza un paso en la investigación y escribe un informe en el que establece la diferencia entre depurar y probar. Según el autor [21], comprobar programas implica dos metas: *“Make sure the program runs”* (Asegurar que el programa funciona) y *“Make sure the program solves de problem”* (Asegurar que el programa resuelve el problema); el primer objetivo respondería a la depuración y el segundo a las pruebas. Al segundo período, que abarcaría desde 1957 hasta 1978, D. Gelperin y B.Hetzel [19] le denominan *“The Demonstration-Oriented Period”*. Durante el mismo se intenta destacar que el propósito de las pruebas consiste en demostrar que los programas son correctos, se contempla la idea de que el software pueda ser probado exhaustivamente de forma que siempre funcione correctamente.

Antes de 1975 a pesar de que las pruebas eran una parte importante en el desarrollo software, se realizaban de forma intuitiva y a medida para un conjunto de métodos. En este año, Goodenough y Gerhart [22] publican el primer artículo dentro de este campo *“Toward a Theory of Test Data Selection”* en el que examinan el papel teórico y práctico de las

pruebas del software dentro del desarrollo. Afirman que el propósito de las pruebas es determinar si un programa contiene algún error, *“un test ideal, por lo tanto, sucede solamente cuando un programa no contiene errores”* [22] y proponen las características que debe cumplir dicho test. Howden W.E [23] en, *“Theoretical and Empirical Studies of Program Testing”*, también habla de las dos aproximaciones: teórica y empírica. La primera se caracteriza por hacer posible el uso de pruebas para demostrar formalmente lo correcto de los programas; en la aproximación empírica se recogen estadísticas que marcan la frecuencia con la que diferentes estrategias de pruebas revelan los errores en una colección de programas. Miller [24] también escribe sobre el propósito de las pruebas: *“los métodos de pruebas se centran en demostrar si una operación es correcta dentro de un sistema software, usando métodos que también aseguren encontrar errores que se hayan producido”*. Dentro de este período, también se desarrolla la primera conferencia formal dedicada a las pruebas software, organizada en 1972 por Hetzel en la Universidad del Norte de Carolina.

El tercer período, al que se refieren D.Gelperin y B.Hetzel [19] como *“The Destruction-Oriented Period”*, abarca desde 1979 hasta aproximadamente 1982. El hito más importante de esta etapa fue desarrollado por G.Myers [25] en 1979, escribe el primer libro sobre pruebas software: *“The Art of Software Testing”* dónde describe los fundamentos necesarios para hacer más efectivo el diseño de pruebas; y que influirá en muchos de sus contemporáneos y autores posteriores. Según el autor, una de las causas principales de que los programas de pruebas fuesen pobres era porque tenían un concepto erróneo de las pruebas. Para Myers cuando se realizan las pruebas, se desea añadir valor a través de las pruebas cuyo significado es incrementar la calidad y fiabilidad del programa; e incrementar la fiabilidad implica encontrar y eliminar errores. Así, define el proceso de pruebas como: *“El proceso de ejecutar un programa con la intención de encontrar errores”* [25]. Además, identifica un conjunto de principios que considera vitales para desarrollar programas de pruebas; lo cuales se enumeran a continuación [25]:

1. Una parte necesaria de un caso de pruebas es tener una definición de la salida o los resultados esperados.
2. Inspeccionar los resultados de cada prueba de forma exhaustiva.
3. Los casos de prueba deben ser escritos tanto para unas condiciones de entrada que sean inválidas o inesperadas como para las que son validas y esperadas.
4. Examinar un programa para ver si no hace lo que se supone que debe hacer es solamente la mitad de la batalla; la otra mitad es ver si el programa hace lo que no se supone que hace.
5. Evitar casos de prueba desechables a menos que el programa sea realmente desechable.
6. No planificar el esfuerzo necesario para las pruebas asumiendo que no se van encontrar errores.

7. La probabilidad de la existencia de más errores en una sección de un programa es proporcional al número de errores ya encontrados en esa sección.
8. Probar es una tarea extremadamente creativa e intelectual.

La publicación de Myers dió lugar a que otros autores comenzasen a escribir, además de artículos, libros sobre el proceso de pruebas. En 1980, Howden W.E [26] publica la primera aproximación teórica hacia como diseñar métodos sistemáticos que puedan ser utilizados para construir pruebas funcionales. Discute la idea de diseñar funciones, las cuales se correspondan con secciones de código documentado con comentarios que describan el efecto de la función. Un año después Howden W.E junto con Miller publica: *“Software Testing and Validation Techniques”* [27], dónde además de tratar el proceso de pruebas abarcan técnicas de validación.

En 1982, D.Gelperin escribe el artículo *“A Software Test Documentation Standard”* en el que justifica por qué es necesaria la documentación en el proceso de pruebas y porque es necesario que ésta cumpla un estándar, así como el propósito de los mismos; lo que constituirá la base para la publicación de diferentes estándares en la siguiente etapa.

El siguiente período se puede establecer entre 1983 y 1987; según D. Gelperin y B.Hetzel [19] es: *“The Evaluation-Oriented Period”*. El Instituto de Ciencias de la Computación y Tecnología del Nacional Bureau of Standards publica *“Guideline for Lifecycle Validation, Verification, and Testing of Computer Software”* [28] en 1983; en la cual especifica una metodología que integra el análisis, revisión y actividades de prueba para realizar la evaluación del producto durante el ciclo de vida del software. Cada fase del ciclo de vida tendrá asociado un conjunto de actividades y productos. La guía proporciona un conjunto de técnicas seleccionadas cuidadosamente que pueden ayudar a asegurar el desarrollo y mantenimiento de software de calidad. En ese mismo año el IEEE Technical Comité on Software Engineering publica: *“Standard for Software Test Documentation”* [29] en el que especifica el contenido y formato de ocho documentos. Esta misma institución publicó en 1986 un estándar para los planes de verificación y validación del software [30] en el que se especifica el contenido y formato que debe cumplir dichos documentos y, en 1987, un estándar para la unidad de pruebas [31] en el que se especifica las fases, actividades, tareas y documentos que forman la unidad de pruebas, poniendo especial énfasis en la necesidad de diseñar el conjunto de pruebas.

Durante esta etapa también se desarrolla una metodología *“Systematic Test and Evaluation Process”* o STEP la cuál proporciona un modelo de procesos y secuencia de actividades y tareas paso a paso para desarrollar pruebas software. Establece tres fases dentro del proceso de pruebas: Planificación, Adquisición y Aseguramiento; teniendo cada una de ellas asignado un conjunto de entradas, actividades y salidas. Hetzel. B. describirá detalladamente la metodología en su libro *“The Complete Guide to Software Testing”* [31].

Beizer B. publica el libro *“SW Testing and Quality Assurance”* [32] que será utilizado como referencia por diferentes investigadores de la materia. Entre otros aspectos, explica

porque se deben desarrollar las pruebas y cuáles son más apropiadas a cada fase de desarrollo del producto.

El último período se puede establecer desde **1988** hasta nuestros días. En ese mismo año, Gelperin y Hetzel [19] publican su artículo *“The growth of Software Testing”* en el que hablan de la trayectoria que ha seguido el proceso de pruebas, y que ha sido y es utilizado como referencia por muchos otros autores. Dos años después, B.Beizer [33] escribe el libro *“Software System Testing Techniques”* el cuál contiene el catálogo más completo de técnicas de prueba. Proporciona una perspectiva general de las pruebas software, enfatizando en los modelos formales para pruebas; así como las razones y metas de las mismas. Ilustra los mejores y peores entornos de pruebas en la industria y permite extraer puntos de referencia a partir de los cuales evaluar y mejorar las prácticas de pruebas. A partir de los **90**, la preocupación de los investigadores se centra en la integración de las actividades de prueba durante el ciclo de vida del software; por lo que las líneas de investigación giran en torno a esta premisa; aunque se sigue trabajando en la generación de datos de prueba. Jean Hartman [34] publica un artículo en el año 2000 en el que propone utilizar el modelado realizado con UML como base para la generación automática de casos de prueba y ejecución durante las pruebas unitarias y de integración.

Además, durante la década de los 90 comienza a surgir un nuevo concepto en torno al proceso de pruebas: modelos de mejora de la calidad en las pruebas que mejoren la calidad del proceso y, por tanto, aumenten la calidad del producto reduciendo su coste. Esta nueva línea de trabajo, no surge de repente ni de forma aislada; sino que se apoya en todas las teorías anteriores y, además, se basa en los modelos de mejora de la calidad del desarrollo software.

Actualmente, existen dos modelos relevantes para la evaluación y mejora de la calidad del desarrollo software: SPICE (actualmente ISO 15504) y CMMI. La 15504 no contempla el proceso de pruebas mientras que la UNE 71044, equivalente a la ISO 12207 en español, si tiene en cuenta procesos relacionados con las pruebas.

Las pruebas constituyen un elemento crítico dentro del proceso de desarrollo del software, del que depende en gran parte la calidad del producto software. Además, el tiempo que requiere el proceso de pruebas es muy elevado, así como los costes asociados al mismo. Si se aumentase la calidad del proceso, se conseguiría aumentar la calidad del producto así como reducir los costes y el tiempo de producción. CMMI y la Norma Española contemplan el proceso de pruebas pero no lo tratan adecuadamente, así para cubrir las deficiencias en el área de pruebas presentado por ambos modelos surgen los modelos de mejora de la calidad del proceso de pruebas, siendo los más importantes el Modelo de Madurez de Pruebas (TMM – Testing Maturity Model), el Proceso de Mejora de Pruebas (TPI-Test Process Improvement) y Aproximación de la Gestión de Pruebas (Tmap – Test Management approach). Estos modelos pueden ser considerados como una extensión de los dos anteriores, por lo que pueden ser utilizados de forma complementaria para obtener una mejora tanto en el proceso de pruebas como en la organización.



## 2.6 Modelos de Referencia de Procesos

### 2.6.1 Primera aproximación: Procesos

Para lograr el entendimiento de que un proceso es necesario, en primer lugar se debe presentar una definición. En la literatura, se puede encontrar una amplia variedad de definiciones de la palabra “proceso”. Por consiguiente, se observa que dentro de esta variedad existen distintas interpretaciones, sean por extensión, cobertura u orientación [35].

Hammer y Champy presentan la siguiente definición de un proceso: “una colección de actividades que toman uno o más tipos de entradas y crea una salida que es de valor para el cliente”. [36]

Zahran en 1998, expuso que un proceso está compuesto por tres aspectos. El primer aspecto es la definición del proceso, que generalmente está representado por un documento que especifica las actividades y procedimientos para el proceso. El segundo aspecto se refiere a la transferencia del conocimiento del proceso para aquellos que lo van a ejecutar. Finalmente, el tercer aspecto son los resultados obtenidos del proceso después de su ejecución. Según Zahran, el enfoque orientado a los procesos permite obtener los siguientes beneficios:

- La sincronización y armonía entre las actividades desempeñadas por cada individuo y los objetivos comunes del equipo.
- La garantía de la consistencia de las actividades y sus resultados.
- El uso de técnicas objetivas de medición del desempeño de cada individuo respecto a las actividades asignadas.
- Debido a la reducción de la dependencia en cada individuo, se consigue que el equipo pueda repetir un determinado proceso.

Enfocándolo a un proceso software, podemos decir que es necesaria una secuencia de pasos para su desarrollo o mantenimiento. En función del ámbito y utilidad, un proceso software define un marco de trabajo tanto en el ámbito técnico como en el ámbito de gestión para la aplicación de métodos, herramientas y personas en la tarea de desarrollo de software. La propia definición del Proceso de Software establece los roles, tareas específicas, métricas y criterios de éxito a aplicar en cada caso. Hay cuatro clases de procesos software: [37]

1. **Procesos informales:** Son procesos en los que no existen un modelo de proceso definido de forma estricta. El proceso utilizado es el elegido por el equipo de desarrollo. Los procesos informales podrían utilizar procesos formales como la gestión de la configuración.
2. **Procesos gestionados.** Se utiliza un modelo de proceso para dirigir el proceso de desarrollo por lo tanto se define los procedimientos, la agenda y la relación entre los procedimientos.

3. **Procesos metodológicos:** Utilizan algún método de desarrollo definido y se benefician de las herramientas CASE para el diseño y análisis.
4. **Procesos de mejora:** Son procesos que tienen objetivos de mejora por tanto disponen de procedimientos para introducir tales mejoras.

Esta clasificación es útil debido a que ayuda a las organizaciones a elegir un proceso de desarrollo apropiado para los diferentes productos. [37]

Cabe destacar a **Watts Humphrey** dentro de los procesos software por incorporar al desarrollo de software los principios de calidad que fueron ideados por Edwards Deming y Philip Crosby, además de propiciar la creación del modelo CMM. El principio básico establecido por Humphrey es la orientación a los procesos vista como la necesidad de contemplar todas las tareas relacionadas con el desarrollo de software además de como un proceso puede ser controlado, medido y mejorado. [38]

Atendiendo a los principios básicos de la calidad como son la prevención y la mejora continua podemos analizar que la calidad se encarga de que no aparezcan disfunciones, y en caso de haberlas se subsanan tan rápido como sea posible. Por ello, es útil el uso de un ciclo de acciones correctivas y preventivas que se centra en la mejora de los procesos. Este ciclo es el llamado “ciclo de Deming” o modelo PDCA, referido a los siguientes cuatro pasos:

1. **Planear (plan):** Definir los objetivos a alcanzar y planificar cómo implementar las acciones.
2. **Hacer (do):** Implementar las acciones correctivas.
3. **Controlar (check):** Verificar que se logre el conjunto de objetivos.
4. **Actuar (act):** Según los resultados obtenidos en el paso anterior, tomar medidas preventivas.

A modo de conclusión, mencionar que para realizar un correcto desarrollo de un proceso es necesario determinar un modelo de referencia debido a que un modelo de este tipo identifica los procesos necesarios que se deben llevar a cabo para asegurar la calidad en un determinado producto.

### 2.6.2 Modelo de Referencia

Actualmente, las organizaciones empeñan un mayor esfuerzo en desarrollar software de calidad debido a que la calidad de los productos está ligada a la calidad de los procesos utilizados en el desarrollo. Para ello es necesario contar con un modelo de referencia el cual se encarga de establecer buenas prácticas para el ciclo de vida del producto software y están orientados a los procesos de gestión y desarrollo de las pruebas. Estos modelos no aseguran la calidad del software pero si no se usan, la calidad será nula.

Por ello, para mejorar la calidad, las organizaciones centran sus esfuerzos en mejorar sus procesos de desarrollo mediante la implantación de modelos de calidad de procesos. El resultado de aplicar estos modelos es óptimo pero está lejos de lograr los cero defectos.

Existen varios modelos encargados de la mejora de los procesos de software como por ejemplo:

- **Modelo de mejora de los procesos de software:** Los *modelos SPI (Software Process Improvement)* se encargan de desarrollar software que sea acorde a la planificación establecida a la vez que mejoran la creación de productos.  
Este modelo es usado en una organización para llegar a un estado de madurez definido, continuo y optimizado. En la actualidad, uno de los modelos más importantes es CMMi del cual se hablará más adelante.
- **Modelo de mejora de los procesos de prueba:** Este modelo debe aplicarse a todos los procesos para obtener un beneficio en la calidad del proyecto.  
Con el fin de obtener productos con calidad, se observó la carencia que había en los procesos destinados a las pruebas. A consecuencia de esta carencia, las pruebas se convertían en un proceso complicado, duradero y con un alto número de recursos. Esto estimuló la creación de modelos orientados a mejorar los procesos de pruebas, con el fin de aumentar la calidad del producto ofertado y de su desarrollo.

A continuación, se definen tres de los modelos más significativos en el ámbito de la calidad software como son el CMMI, TMM y TMMI.

### **CMMI: Modelo de capacidad y madurez integrado**

A finales del siglo XX, un estudio encargado por el departamento de defensa de EE.UU. (DoD) reveló que los proyectos pactados por 28 meses sufrían retrasos de otros 20 y, que un proyecto de 4 años no terminaba en menos de 7 [39]. La necesidad de solventar este problema provocó el nacimiento de CMM (Capability Maturity Model) impulsado por Watts S. Humphrey y cuyo propósito es ser un modelo de calidad de desarrollo software. Fue el primer gran modelo de calidad que englobaría las distintas metodologías surgidas hasta la fecha, y ofrecería al usuario una evaluación sobre la madurez de sus procesos.

Años más tarde se desarrolló CMMI [40] que es un derivado de CMM e incorpora mejoras basadas en la experiencia. Se define CMMI como modelo de referencia que engloba el desarrollo y el mantenimiento tanto de productos como de servicios software. En él, se desarrollan prácticas que abarcan la gestión del proyecto, la gestión de procesos, la ingeniería de sistemas, la ingeniería del software, la ingeniería del hardware y otros procesos de soporte utilizados en el desarrollo y en el mantenimiento. Gracias a organizaciones como el DoD (Diseño, Organización y Distribución), National Defense Industrial Association (NDIA), y del Software Engineering Institute (SEI), se ha convertido en un modelo ampliamente utilizado en más de 30 países distintos tal y como señala la Web [41] del propio SEI. CMMI ofrece una mejora de procesos a partir de dos representaciones

distintas: representación por etapas y representación continua, definiendo para cada una de ellas un conjunto de niveles que engloban distintas áreas de proceso software.

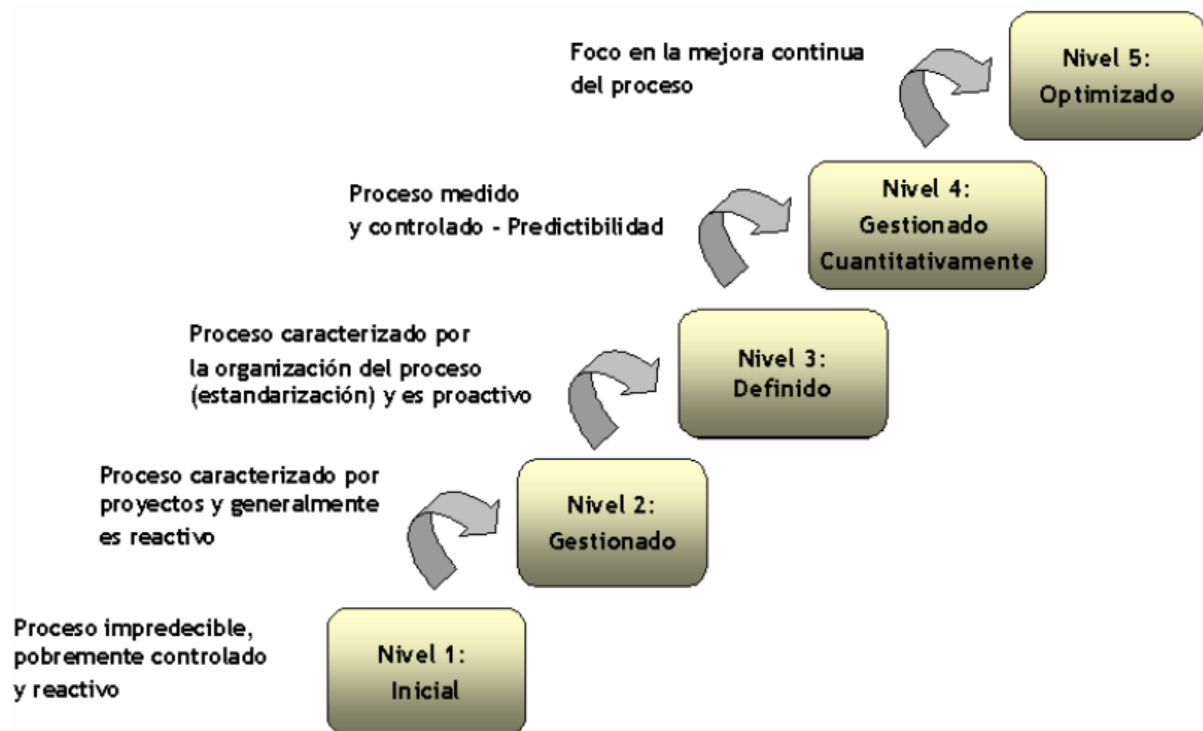


Figura 5: Niveles de Madurez del modelo CMMI

	REPRESENTACIÓN CONTINUA	REPRESENTACIÓN POR ETAPAS
NIVEL	NIVEL DE CAPACIDAD	NIVEL DE MADUREZ
0	Incompleto	N/A
1	Ejecutado	Inicial
2	Gestionado	Gestionado
3	Definido	Definido
4	Gestionado cuantitativamente	Gestionado cuantitativamente
5	Optimizado	Optimizado

Figura 6: Comparación de las representaciones del Modelo CMMI

## TMM: Modelo de madurez de las pruebas

El modelo de madurez de las pruebas (Testing Maturity Model, TMM), fue desarrollado por el Instituto de Tecnología de Illinois, con el propósito de servir como una guía para realizar una mejora del proceso de pruebas además de servir como complemento a CMM. TMM al igual que CMM utiliza el concepto de niveles de madurez para la evaluación y la mejora de los procesos.

TMM utiliza el modelo de la evolución de las pruebas de (Gelperin y Hetzel, 1988). [42] Este modelo refleja las fases que se deben desarrollar a la hora de realizar las pruebas en una organización, las cuales van desde la detección de defectos hasta la prevención de los mismos. Utiliza conceptos fundamentales como: dirección, medición, supervisión y eficacia con el fin de probar la madurez de los procesos [43] y poder implementar los cinco niveles de madurez. Estos niveles son: Inicial, Definición, Integración, Gestionado/Medido y Optimización, niveles que permiten mejorar la eficiencia en las pruebas permitiendo a las organizaciones migrar sus procesos inmaduros a procesos probados y maduros.

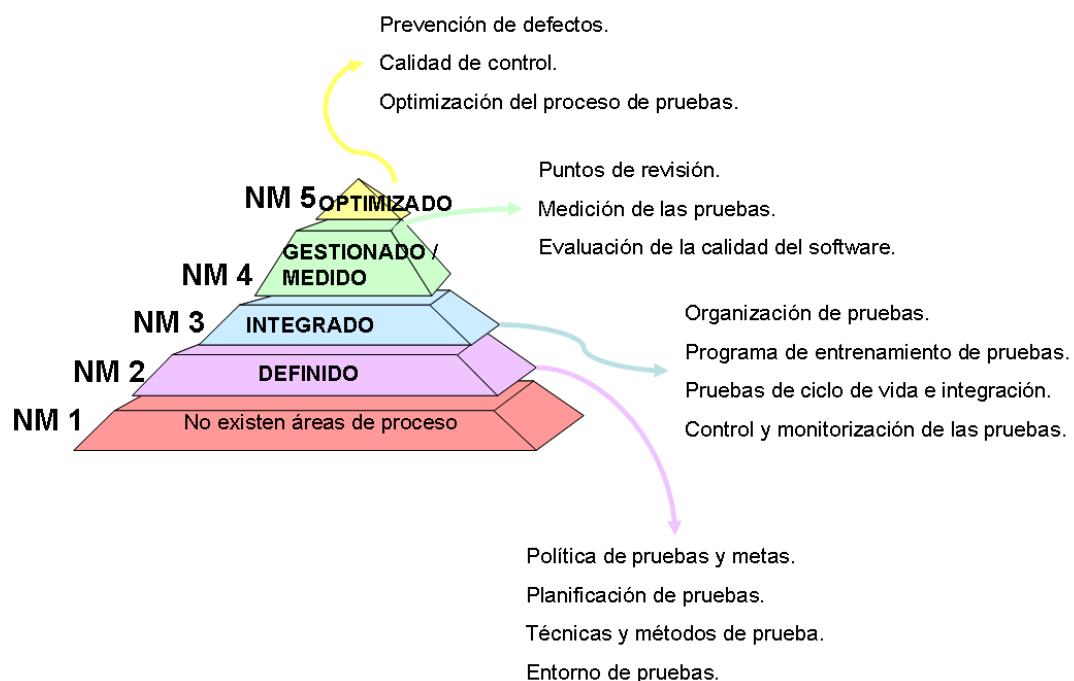


Figura 7: Modelo TMM

## TMMi: Modelo de madurez de las pruebas integrado

Testing maturity model integrated (TMMi) [44] es la evolución de TMM, desarrollada por “TMMi foundation” para mejorar el proceso de pruebas. Dicho modelo surgió como respuesta y complemento de CMMi, ya que su enfoque al ser más amplio y estructurado, logra un impacto favorable sobre la calidad del producto.

La estructura de TMMi se basa principalmente en la estructura de CMMi, presentando una mejora en los procesos relativos a los procesos de prueba y definiendo también niveles

de madurez [44] con la diferencia de que TMMi hasta el momento solo ha desarrollado la representación por etapas.

Los niveles de madurez de TMMi se encuentran divididos en un conjunto de áreas de proceso. Éstas son un conjunto de actividades que una organización necesita realizar para alcanzar un determinado nivel de madurez. Al igual que en CMMI, un área de proceso se consigue gracias a logro de una serie de prácticas claramente diferenciadas, clasificadas como componentes esperados y requeridos.

Los “**componentes esperados**” describen qué debe implementar una organización para alcanzar un componente requerido. Se conocen como componentes esperados a las prácticas específicas, que son descripciones de una actividad importante para conseguir un objetivo específico; y a las prácticas genéricas, las cuales describen actividades importantes para conseguir un objetivo genérico.

Los “**componentes requeridos**” describen qué debe alcanzar una organización para satisfacer un área de proceso. Se distinguen los objetivos específicos, que describen características necesarias para satisfacer un determinado área de proceso; y los objetivos genéricos, describen características comunes a distintas áreas de proceso. En la figura 7 se muestra la estructura de dicho modelo. [15]



Figura 8: Estructura TMMi

## 2.7 Externalización de las Pruebas Software

### 2.7.1 Outsourcing

Outsourcing es una práctica que data desde el inicio de la era moderna y consiste básicamente en la contratación externa de recursos mientras la organización se dedica exclusivamente a la actividad de su proyecto. Este concepto no es nuevo, ya que muchas compañías competitivas lo realizaban como una estrategia de negocios. [45]

El concepto de Outsourcing comienza a ganar credibilidad al inicio de la década de los 70's enfocado, sobre todo, a las áreas de información tecnológica de las empresas. Las primeras empresas en implementar modelos de Outsourcing fueron gigantes como EDS, Arthur Andersen y Price Waterhouse.

Es un término creado en 1980 para describir la creciente tendencia de grandes compañías a transferir sus sistemas de información a proveedores. En 1998, el Outsourcing alcanzó una cifra de negocio a nivel mundial de cien mil millones de dólares. [45]

Básicamente se trata de una modalidad según la cual, determinadas organizaciones, grupos o personas ajenas a la organización son contratadas para hacerse cargo de "parte del proyecto" o de un servicio puntual dentro de ella. La compañía delega la gerencia y la operación de uno de sus procesos a un prestador externo (Outsourcer o Suministrador) con el fin de agilizarlo, optimizar su calidad y/o reducir sus costes.

Para ello, es preciso pasar de un enfoque de abastecimiento tradicional que consiste en un conjunto de actividades que permite identificar y adquirir los bienes y servicios que la organización requiere para su operación interna o externa, a una visión estratégica enfocada a aumentar el valor y la calidad de los productos. Para aumentar la calidad del producto y garantizar que su proceso de desarrollo es adecuado será necesario aplicar los modelos de calidad que han sido descritos en el apartado anterior.

La principal ventaja del Outsourcing es que la organización se preocupa exclusivamente por definir la funcionalidad de las diferentes áreas, dejando que la empresa de Outsourcing se ocupe de decisiones de tipo tecnológico, manejo del proyecto, implementación, administración y operación de la infraestructura. Otras posibles ventajas o beneficios del proceso de Outsourcing son: [45]

- Permite a la organización responder con rapidez a los cambios del entorno.
- Incremento en los puntos fuertes de la organización.
- Construye una larga ventaja competitiva sostenida mediante un cambio de reglas y un mayor alcance de la organización.
- Incrementa el compromiso hacia un tipo específico de tecnología que ayuda a mejorar el tiempo de entrega y la calidad de la información para las decisiones críticas.
- Permite a la empresa poseer lo mejor de la tecnología sin la necesidad de entrenar al personal de la organización para manejarla.
- Permite disponer de servicios de información de forma rápida considerando las presiones competitivas.
- Aplicación del talento y los recursos de la organización a las áreas claves.
- Aumento de la flexibilidad de la organización y disminución de sus costes fijos.

Sin embargo, como en todo proceso existen aspectos negativos y el Outsourcing no queda exento de esta realidad. Se pueden mencionar las siguientes desventajas respecto al Outsourcing: [45]

- Estancamiento en lo referente a la innovación por parte del suministrador externo.
- La organización pierde contacto con las nuevas tecnologías que ofrecen oportunidades para innovar los productos y procesos.
- El suministrador externo al tener conocimiento del producto en cuestión tiene la posibilidad de usarlos para empezar un proyecto propio y pasar de suministrador a competidor.
- El coste ahorrado con el uso de Outsourcing puede que no sea el esperado.
- Alto coste en el cambio de suministrador en caso de que el seleccionado no resulte satisfactorio.
- Reducción de beneficios
- Pérdida de control sobre la producción.

Destacar que uno de los riesgos más importantes del Outsourcing es que el proveedor seleccionado no tenga las capacidades para cumplir con los objetivos y estándares que la empresa requiere. De aquí la necesidad de disponer de un modelo de procesos robusto y bien definido que pueda soportar diferentes niveles de servicio, de modo que este puede adaptarse tanto a organizaciones que requiera de un mínimo de requisitos hasta organizaciones que necesiten cumplir objetivos más exigentes.

En conclusión, a pesar de los diferentes inconvenientes que presenta, la externalización en una práctica beneficiosa para una organización debido a que se libera de carga laboral y por tanto se ahorra tiempo y esfuerzo personal en tareas que son consideradas menores.

### 2.7.2 Externalizar pruebas software

En la actualidad se está comenzando a externalizar las pruebas software. Como se apunta en la introducción, existen organizaciones que no pueden hacer frente al coste que supone definir e implementar los procesos de Verificación y Validación que permiten incrementar la calidad de los productos. Es por ello que prefieren subcontratar este tipo de actividades a organizaciones expertas en ello, que dispongan de los recursos, procesos e infraestructura adecuada.

Las organizaciones subcontratadas y por tanto encargadas de realizar actividades de Verificación y Validación, entre las que se encuentran las pruebas software, llevaran a cabo actividades como: [46]

- Administración de Proyectos de Verificación y Validación.
- Ejecución de las Tareas de Verificación y Validación.
- Revisiones Formales de Documentación.
- Revisiones de Código.
- Pruebas de Validación.
- Pruebas Unitarias y de Integración.
- Pruebas Funcionales y de Aceptación.



- Administración de Anomalías, Análisis y Prevención de Defectos.

Además externalizar las pruebas software permite verificar las condiciones del desarrollo, incluida la seguridad. El problema que presenta la verificación de los requisitos de seguridad radica en la necesidad de la intervención de personal especializado. A continuación, se proponen diferentes medidas a adoptar para resolver dicho problema: [58]

- **Pruebas realizadas por el propio proveedor.** Es importante requerir los detalles de las pruebas. Puede solicitarse que el equipo de pruebas sea diferente al equipo de desarrollo para cumplir así con el principio de independencia.
- **Pruebas realizadas por la empresa contratante.** En este caso, se diferencian dos tipos, según quién las realice:
  - **Revisión interna,** si la empresa contratante cuenta con un equipo de seguridad. Otra posibilidad consiste en dar formación al equipo de pruebas sobre seguridad; el uso de herramientas automáticas para la detección de vulnerabilidades en modo caja negra así como saber analizar el código. Si se opta por esta alternativa, se recomienda requerir los servicios de una empresa externa en una fase más avanzada, para así contrastar resultados.
  - **Revisión subcontratada a un proveedor especializado.** El primer paso es decidir la modalidad de la revisión: caja blanca o caja negra. La revisión en caja blanca facilita una revisión en amplitud (visibilidad completa y posibilidad de detectar todas las amenazas), mientras que en caja negra se puede simular un ataque real (se trabaja más en profundidad que en amplitud).

## **2.8 Conclusiones**

A lo largo de este capítulo, hemos abordado diferentes áreas de investigación que se encuentran relacionadas con el modelo propuesto.

En primer lugar, hablamos de calidad como cualidad indispensable en los productos software. El objetivo perseguido por el modelo que se propone, será conseguir mejorar la calidad de los productos.

A continuación, hablamos de Verificación y Validación como actividades indispensables para obtener productos libres de faltas o errores. Estas actividades, que forman parte del proceso de SQA, se centran en identificar defectos en los productos software mediante técnicas estáticas, como revisiones, auditorías o inspecciones.

Dentro de las técnicas dinámicas, encontramos las pruebas de software, como elemento esencial para depurar los errores en codificación. El modelo propuesto contemplará las actividades de verificación y validación necesarias para eliminar los fallos que puedan tener los productos software.

Dado que se va a desarrollar un modelo de referencia de procesos, se ha considerado adecuado incluir una definición de los mismos y resaltar aquellos que son de mayor importancia en la actualidad.

Por último, se ha hablado de la externalización y de la importancia de la misma en la actualidad, como alternativa no sólo a los procesos de desarrollo sino también a las actividades de prueba. De modo que el modelo desarrollado pueda ser implantado también en aquellas organizaciones que deseen basar su actividad de negocio en proveer servicios relacionados con la verificación y validación del software.

## CAPÍTULO 3

# PROPUESTA DE MODELO DE REFERENCIA PROCESOS

### **3. PROPUESTA DE MODELO DE REFERENCIA DE PROCESOS**

#### **3.1 Proceso de definición de las fases de desarrollo**

En este apartado se explica cómo se han llevado a cabo cada una de las fases de desarrollo realizadas en el proyecto. Para ello se describe que tareas ha sido necesario aplicar en cada una de las fases.

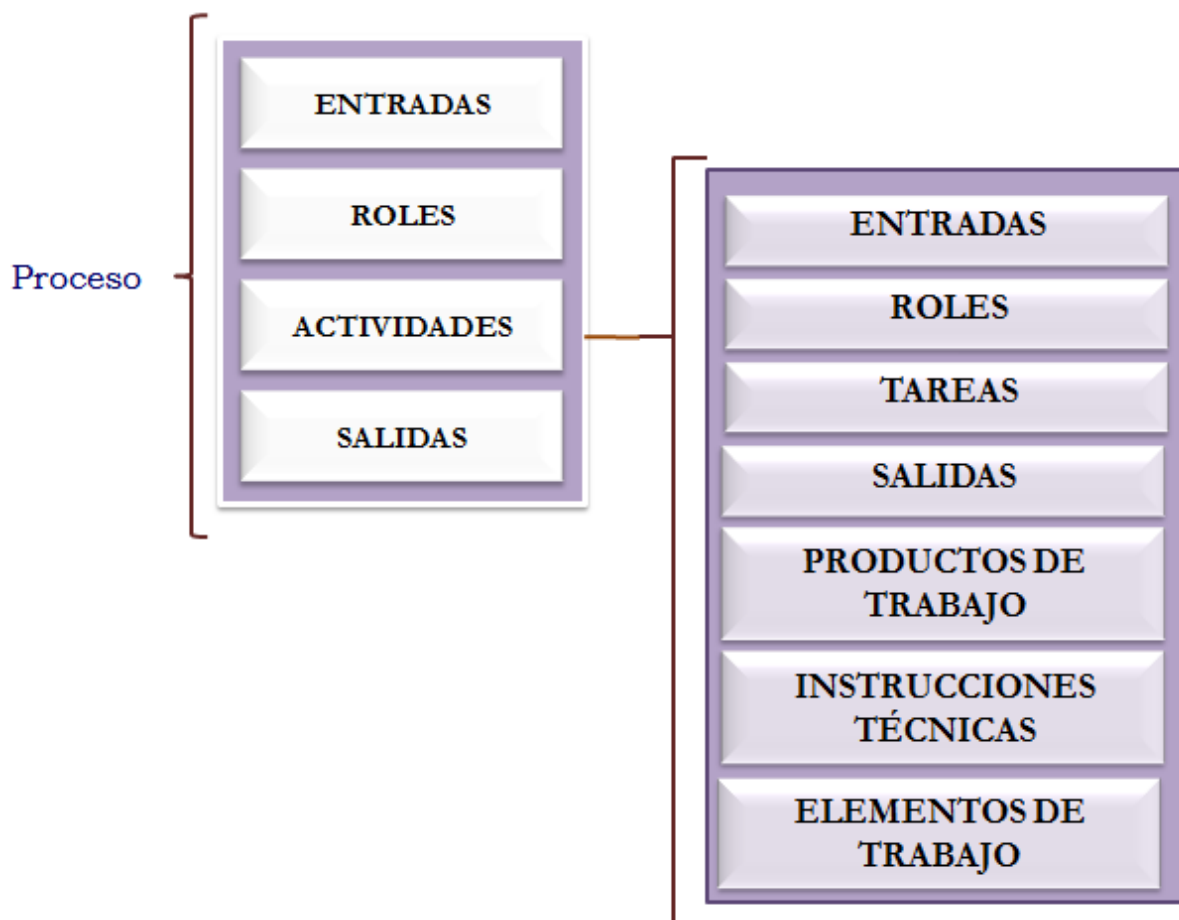
- **Fase 1: Identificar documentación relevante:** En esta fase se han realizado las siguientes acciones:
  - Búsquedas en internet sobre calidad, aseguramiento de la calidad y gestión de la configuración en los productos software.
  - Lectura de libros de ingeniería de software sobre pruebas software, outsourcing, verificación y validación.
  - Recibir información de los tutores del proyecto acerca de que es un modelo de referencia y un modelo de procesos.
- **Fase 2: Analizar documentación:** En esta fase se han realizado las siguientes acciones:
  - Lectura detallada de la información para saber si era información útil o no en mi proyecto.
  - Asesoramiento por parte de los tutores del proyecto acerca de que información iba a ser tratada en el proyecto.
- **Fase 3: Diseñar los procesos:** En esta fase se han realizado las siguientes acciones:
  - Documento que recoge los conceptos encargados de determinar que un producto es de calidad.
  - Traducir los conceptos que aseguran la calidad de un producto a procesos.
  - Esquema que contiene los procesos encargados de asegurar que un producto es de alta calidad.
- **Fase 4: Definir los procesos:** En esta fase se han realizado las siguientes acciones:
  - Identificar las actividades que componen el proceso.
  - Identificar las tareas que componen cada una de las actividades del proceso.
  - Identificar las entradas y salidas del proceso.
  - Documento por cada proceso que recoja toda la información anterior.
- **Fase 5: Seguimiento de los procesos:**
  - Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos.

▪ **Fase 6: Evaluar el proyecto:**

- Lectura del proyecto por parte de los tutores.
- Lectura del proyecto por parte del autor del mismo.

### **3.2 Especificación de los procesos**

A continuación, se explica cómo han sido definidos los procesos que van a formar parte del Modelo de Referencia propuesto en este PFC. La Figura 9 ilustra que componentes integran un proceso.



**Figura 9: Especificación de los procesos**

Para llevar a cabo la especificación de los procesos se han realizado las siguientes acciones:

1. Obtener una visión general del proceso para saber en qué consiste el proceso y cuando sucede. Además, a través de la visión general sabremos por qué es importante el proceso y cuáles son sus principales beneficios.
2. Identificar el conjunto de actividades que forman el proceso y que por tanto permiten alcanzar el objetivo de dicho proceso así como saber cuáles son sus entradas, salidas y roles.

3. Conocer por cada actividad que forma parte del proceso en qué consiste, quiénes participan en dicha actividad, qué entradas necesita para llevarse a cabo así como las salidas que genera tras ejecutarse. También es necesario tener conocimiento acerca de cuáles son los productos de trabajo que utiliza, qué elementos de trabajo están involucrados en la actividad y qué instrucciones técnicas pueden ser útiles de usar en dicha actividad.
4. Identificar las entradas que necesita el proceso para ejecutarse así como que tipo de salidas son las que se generan tras su ejecución.
5. Conocer cuáles son los roles que intervienen en la realización del proceso.

A partir de esta estructura, se procedió a elaborar la documentación de cada uno de los procesos en papel, *ver anexo 4*, priorizando los documentos de cada proceso, ya que a partir de ellos se puede descender jerárquicamente hacia actividades y tareas, y a los respectivos productos de trabajo e instrucciones técnicas. Para redactar esta documentación, se dispuso de documentación personal y de aportaciones de la tutora del proyecto, de modo que se conformó una base de conocimiento acerca del Proceso Software que fue revisado hasta generar la documentación final que contendría la guía electrónica de procesos asociada al proyecto.

### **3.3 Modelo de Referencia de Procesos**

Debido a los numerosos defectos existentes en los productos software ocasionados por la falta de calidad, es necesario realizar sobre los productos software una serie de procesos de Verificación y Validación que tengan como objetivo prevenir y eliminar todos los defectos que se detecten en ellos, obteniendo como salida productos de máxima calidad y totalmente útiles para el usuario final. Para ello, se propone seguir el siguiente modelo de referencia de procesos de Verificación y Validación que se describe a continuación. Este modelo describe de forma ordenada cuál es la manera de llevar a cabo los procesos que garantizan la calidad en un producto además de conocer cuáles son las actividades que se efectúan dentro de cada proceso. El resultado de llevar a cabo dicho modelo es la obtención de un producto que posea la mejor calidad y que satisfaga todas las necesidades del cliente.

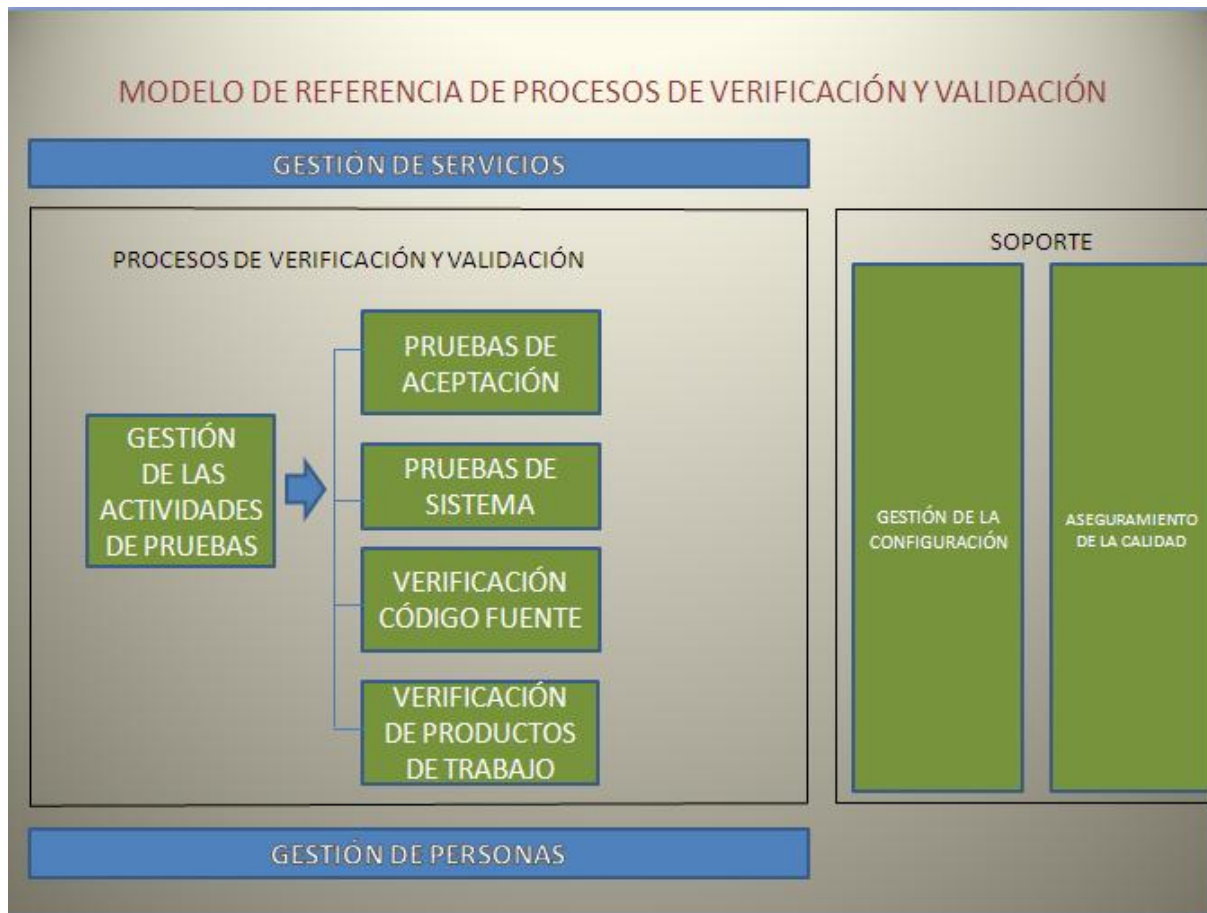


Figura 10: Modelo de Referencia

Este modelo tiene como objetivo principal mejorar la calidad de los productos software desarrollados por una Organización. Para ello, se comenzará por determinar cuáles son los problemas que existen en la Organización así como los pasos a dar para solucionar dichos problemas. De esto se encarga el proceso **Gestión de las Actividades de Pruebas**.









Después de saber cuáles son las medidas a tomar, la Organización debe llevar a cabo los procesos de **Pruebas de Aceptación**, **Pruebas de Sistema**, **Verificación Código Fuente** y **Verificación de Productos de Trabajo** para conseguir que los productos finales sean lo más correctos y adecuados posible. Además, es necesario llevar a cabo los procesos de **Gestión de la Configuración** y **Aseguramiento de la Calidad** ya que sirven de soporte a los procesos de Verificación y Validación garantizando que dichos productos verdaderamente son apropiados.

Aclarar que cada uno de los procesos que forman parte del conjunto de procesos de Verificación y Validación se pueden realizar de forma independiente o en un subconjunto en función de las necesidades que tenga la Organización que va a usar dicho modelo.

Para describir desde otra perspectiva en qué consiste cada uno de los procesos que encontramos en el Modelo de Referencia se va a utilizar el metamodelo SPEM (Software Process Engineering Metamodel). Este modelo permite representar los procesos ya que es

un estándar de la OMG (Object management group) cuyo objetivo principal es proporcionar un marco formal para la definición de procesos de desarrollo de sistemas y de software así como para la definición y descripción de todos los elementos que los componen. [47]

Otra de las razones por las cuales se ha utilizado SPEM es porque además de como metamodelo se define como un perfil UML, siendo los siguientes iconos los usados en la representación gráfica de los procesos: [48]

ICONO	SIGNIFICADO
	Icono que representa un <b>proceso</b> .
	Icono que representa cada una de las <b>entradas</b> y <b>salidas</b> que hay en el proceso y en cada de una de las actividades de dicho proceso.
	Icono que representa cada una de las <b>actividades</b> por las que tiene que pasar un proceso.
	Icono que indica quienes son los <b>responsables de un proceso</b> y de las actividades que componen dicho proceso.
	Icono que representa el <b>orden de las tareas</b> a realizar dentro de una determinada actividad.
	Icono que representa los <b>elementos de trabajo</b> que existen dentro de una actividad.
	Icono para representar las <b>instrucciones técnicas</b> que existen dentro de una actividad.
	Icono para representar los <b>productos de trabajo</b> que existen dentro de una actividad.

A continuación, se describen cada uno de los procesos que forman el modelo de referencia utilizando el metamodelo descrito anteriormente, salvo el proceso Evaluación del Proceso de Pruebas y el proceso Gestión de Servicios del cual podemos decir que contiene los procesos necesarios para gestionar los servicios que proponen estos procesos. El motivo de que estos procesos no hayan sido descritos es debido a que quedan fuera del alcance de este proyecto por ser procesos muy amplios siendo la evaluación un área muy extensa. Por tanto, hemos centrado este proyecto de manera más específica en el proceso de Verificación y Validación.

Por último destacar que si se decidiera desarrollar el proceso Gestión de Servicios, el modelo de referencia presentado se podría usar para llevarlo a cabo por ser un modelo útil tanto para las actividades internas de la organización como para las actividades externas.



### ➤ Proceso: Gestión de las Actividades de Pruebas

El objetivo principal de este proceso consiste en implantar y gestionar un proceso de pruebas disciplinado y bien definido en la organización. Esto implica realizar el ciclo de vida de las pruebas desde su identificación hasta su ejecución, análisis y resultados.

La figura 11 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo.

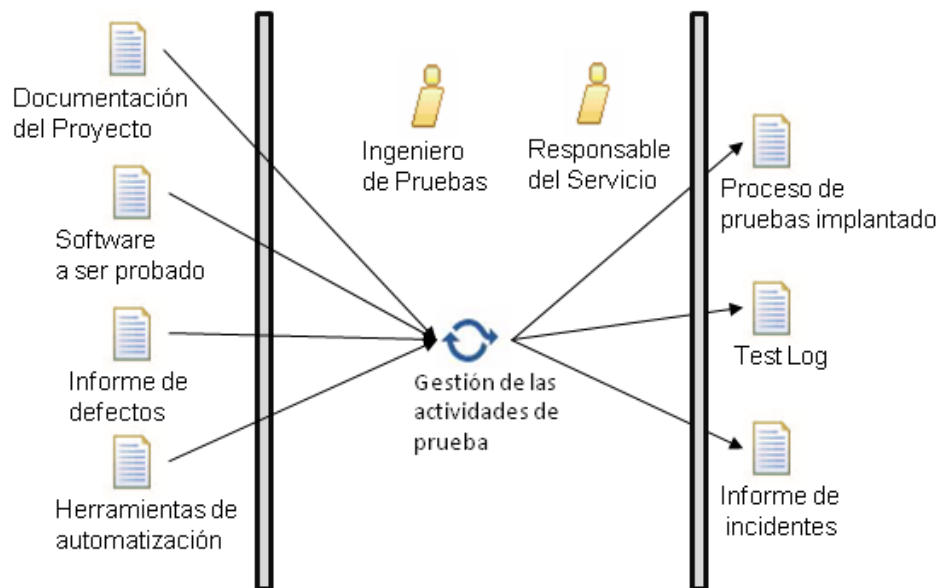


Figura 11: Proceso Gestión de las Actividades de Pruebas

En este proceso podemos distinguir una serie de etapas que abarca desde establecer los objetivos y políticas necesarios hasta la evaluación de las pruebas. En la figura 12, se muestran las distintas actividades que definen el proceso.

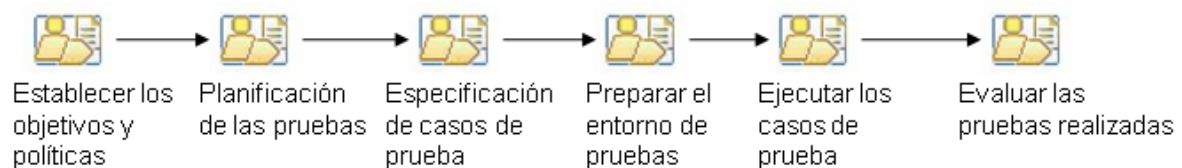
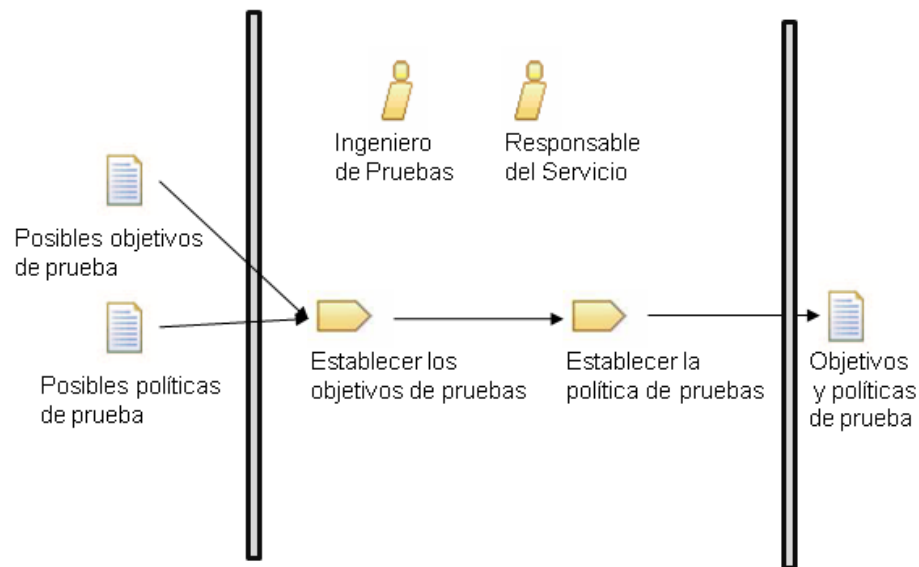


Figura 12: Actividades del proceso Gestión de las Actividades de Pruebas

#### **Actividad1: Establecer los objetivos y políticas**

El objetivo de esta actividad es identificar y determinar los objetivos y políticas asociados a las pruebas en base a los posibles objetivos y políticas de prueba. Para realizar esta actividad se recomienda identificar y definir los objetivos que se pretenden alcanzar con la realización de las pruebas y las políticas que se van a usar. Estas deben soportar el alcance de los objetivos proporcionando una guía que indique cómo se deben desarrollar las pruebas.

Finalmente, se obtendrá una lista que contiene los objetivos y políticas de prueba alcanzados.



**Figura 13: Actividad1: Establecer los objetivos y políticas**

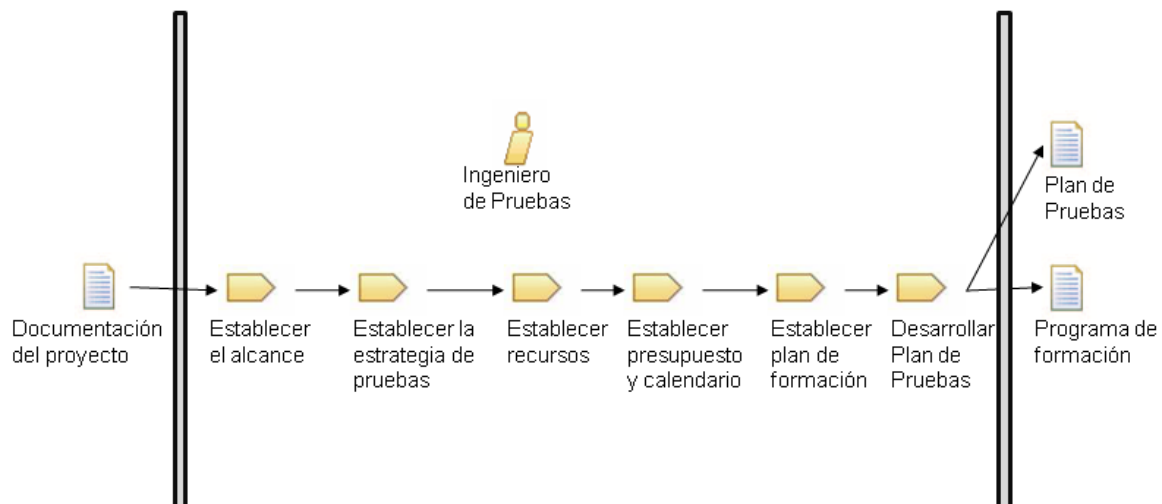
## **Actividad 2: Planificación de las pruebas**

El objetivo de esta actividad es desarrollar el Plan de Pruebas para conseguir que un proceso sea controlable y por tanto las pruebas tengan éxito. La planificación debe comenzar al inicio del proyecto para poder realizar una estimación del tiempo, presupuesto y recursos real, tomando como base la documentación del proyecto.

Para realizar esta actividad se recomienda definir el alcance de las pruebas así como la estrategia de pruebas que se vaya a aplicar; para lo cual habrá que analizar los riesgos del proyecto, priorizar las características que se vayan a probar y conocer la cobertura de la que se puede disponer teniendo en cuenta las restricciones de tiempo o económicas. Además, se tendrá que identificar los recursos tanto personales como materiales para poder asignar las distintas responsabilidades a los miembros que vayan a realizar las actividades de pruebas. Estos miembros forman un equipo de pruebas y recibirán una formación que proporcione el conocimiento, habilidades y capacidades necesarias para desarrollar las actividades de pruebas correctamente.

También en esta actividad es necesario determinar el presupuesto y el calendario a usar; los cuales estarán basados en la estimación del tiempo y esfuerzo necesario para desarrollar las actividades de prueba.

Por último, se realiza el Plan de Pruebas usando la información anterior obteniendo como resultado el documento de dicho plan además del documento que contiene el programa de formación.



**Figura 14: Actividad 2: Planificación de las pruebas**

### **Actividad 3: Especificación de casos de prueba**

El objetivo de esta actividad es recopilar toda la información necesaria para realizar una ejecución efectiva de las pruebas. La elaboración del Plan de Pruebas no es suficiente al no tener en cuenta detalles importantes en la ejecución de las pruebas como puede ser cuales son las entradas requeridas, las salidas esperadas o el procedimiento que se debe seguir en cada una de ellas.

La especificación de los casos de prueba se lleva a cabo después de que los requisitos son aprobados y no cuando el software ha sido desarrollado. De esta forma se definen casos de prueba orientados a los requisitos y se debe tener en cuenta que si se produce alguna modificación de los requisitos o se añade alguno nuevo, será necesaria la revisión de la especificación de los casos de prueba.

Para realizar esta actividad, se parte de la documentación del proyecto así como del Plan de pruebas que ha sido obtenido en la actividad anterior. Se comienza por especificar el diseño de pruebas haciendo uso de aquellas características que van a ser probadas y de las pruebas asociadas a dichas características.

Esta especificación del diseño establecerá una lista con los casos de prueba definidos y por último se obtendrá una lista que contenga la especificación de los procedimientos de prueba en la cual se determinan los pasos que se deben realizar durante la ejecución de los casos de prueba.

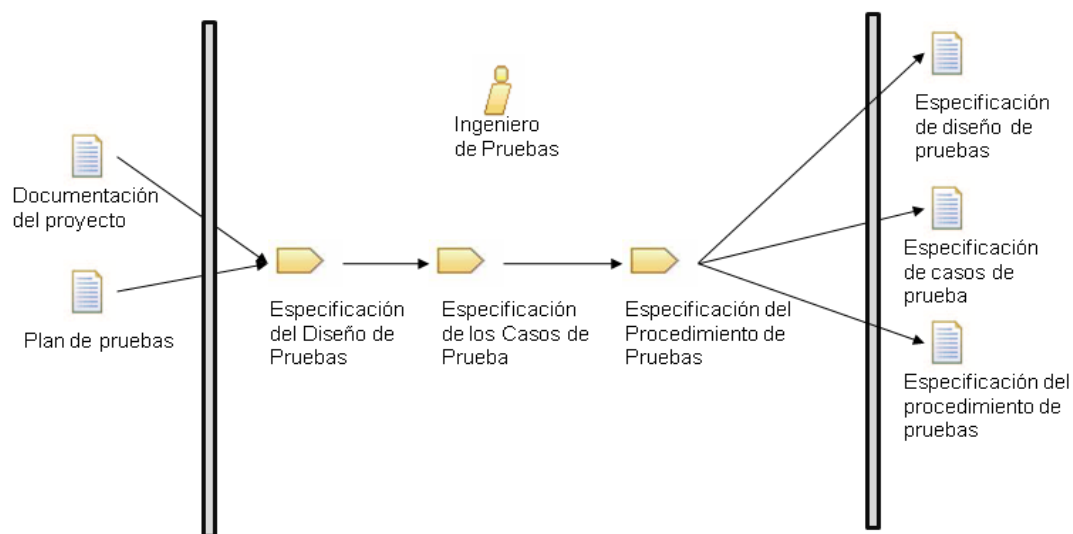


Figura 15: Actividad 3: Especificación de casos de prueba

#### Actividad 4: Preparar el entorno de pruebas

El objetivo de esta actividad es llevar a cabo la preparación del entorno necesario para ejecutar el conjunto de pruebas que ha sido especificado, en base a las especificaciones de las pruebas y al Plan de Pruebas y cuyo propósito es evaluar que el producto funciona correctamente.

Para realizar esta actividad, se recomienda preparar el hardware, el software, librerías y datos para poder llevar a cabo la realización de las pruebas así como generar los datos que se consideren necesarios en la ejecución de los casos de prueba. Cada caso de prueba tendrá un código asociado que debe ser generado y ejecutado para que cuando se lleve a cabo la ejecución esté disponible.

Por último se debe crear las bases de datos o ficheros que se encarguen de soportar el almacenamiento y recuperación de los datos manejados por el sistema.

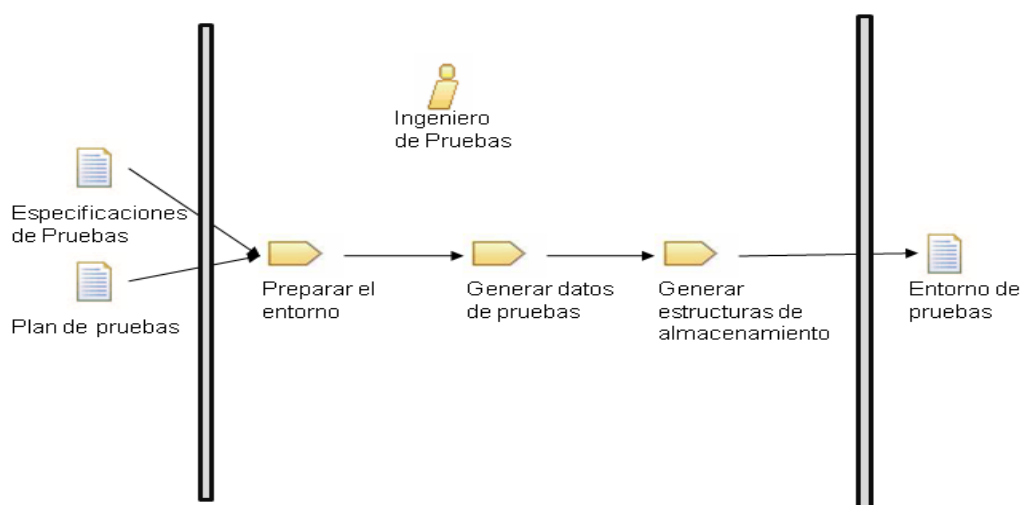


Figura 16: Actividad 4: Preparar el entorno de pruebas

### Actividad 5: Ejecutar los casos de prueba

El objetivo de esta actividad es realizar la ejecución de los casos de prueba generados, tomando como base las especificaciones de las pruebas, el Plan de Proyecto de Pruebas y el entorno de pruebas que ha sido generado.

Para realizar esta actividad, se recomienda determinar mediante el uso de la ficha de pruebas de sistema qué casos de prueba son los que van a ser automatizados para después utilizar los script generados y respetar la secuencia en la que se deben ejecutar las pruebas.

Finalmente, se obtiene un documento que contiene los resultados obtenidos tras la ejecución de las pruebas, además de un Test Log con los detalles relevantes de la ejecución.

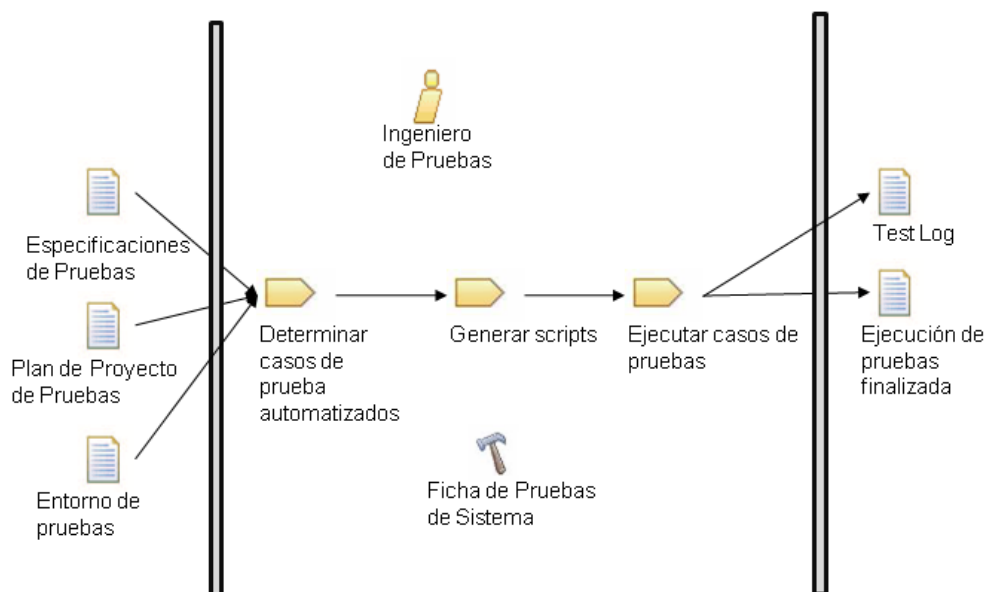


Figura 17: Actividad 5: Ejecutar los casos de prueba

### Actividad 6: Evaluar las pruebas realizadas

El objetivo de esta actividad es analizar el resultado obtenido de la ejecución de las pruebas de sistema y poder evaluar en qué medida han sido cumplidos los requisitos. Para ello, se comparará el resultado obtenido durante la ejecución con el resultado que se espera, tomando como base la Ficha de Pruebas y el Plan de Proyecto de Pruebas.

Para comenzar esta actividad, se recomienda comenzar por comprobar si se ha producido algún fallo en la ejecución y en caso de ser afirmativo el defecto deberá ser encontrado para depurarlo y volver a repetir las pruebas. Una vez repetidas las pruebas, se comprueba que la ejecución ha sido satisfactoria y en caso contrario se debe determinar cuáles son las condiciones anormales que están provocando que la ejecución no se lleve a cabo.

Por último, se evalúan los resultados obtenidos y se debe entregar al cliente un informe de resultados si la evaluación es positiva. En caso contrario, se obtiene un informe de

incidentes que será entregado al cliente y al Jefe de Proyecto, que se encargará de tomar las acciones que estime necesarias para su resolución.

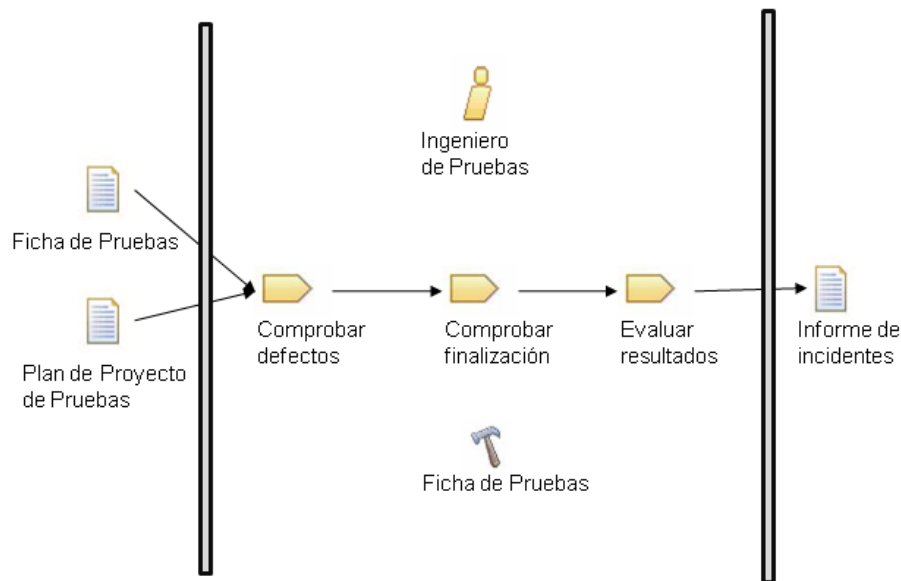


Figura 18: Actividad 6: Evaluar las pruebas realizadas

### ➤ Proceso: Pruebas de Aceptación

El objetivo principal del proceso de Pruebas de Aceptación es llevar a cabo el conjunto de pruebas sobre el producto con el propósito de verificar que satisface los criterios de aceptación, las necesidades y expectativas del cliente.

La figura 19 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo.

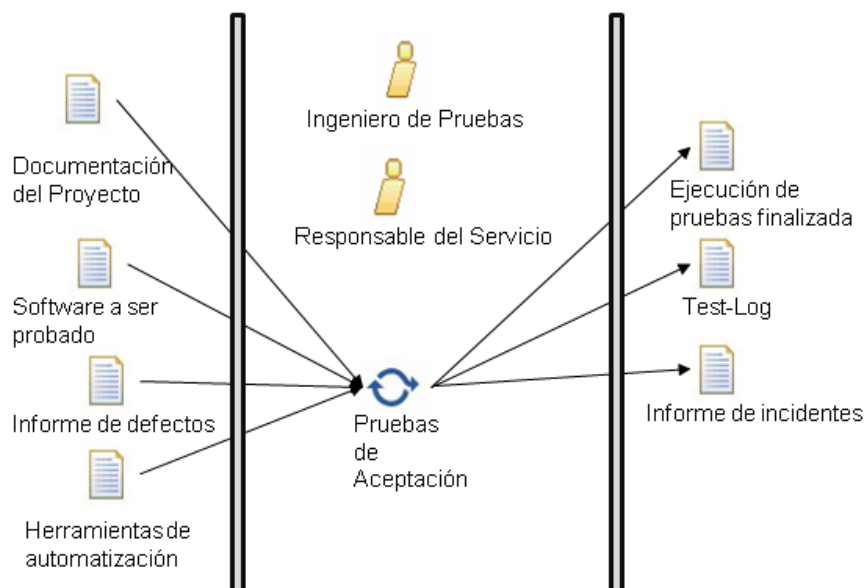
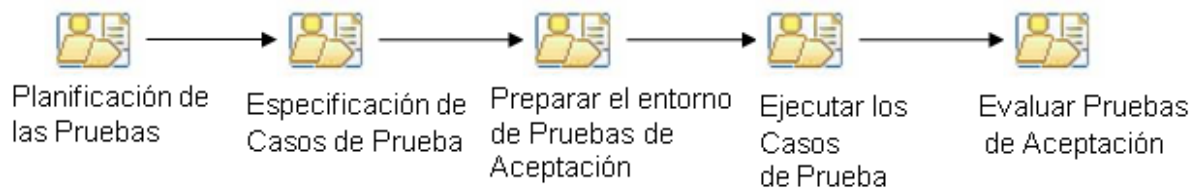


Figura 19: Proceso Pruebas de Aceptación

En la gestión de un proceso de pruebas podemos distinguir una serie de etapas que abarca desde la planificación de las pruebas hasta llegar a su evaluación. En la figura 20, se muestran las distintas actividades que definen el proceso.



**Figura 20: Actividades del proceso Pruebas de Aceptación**

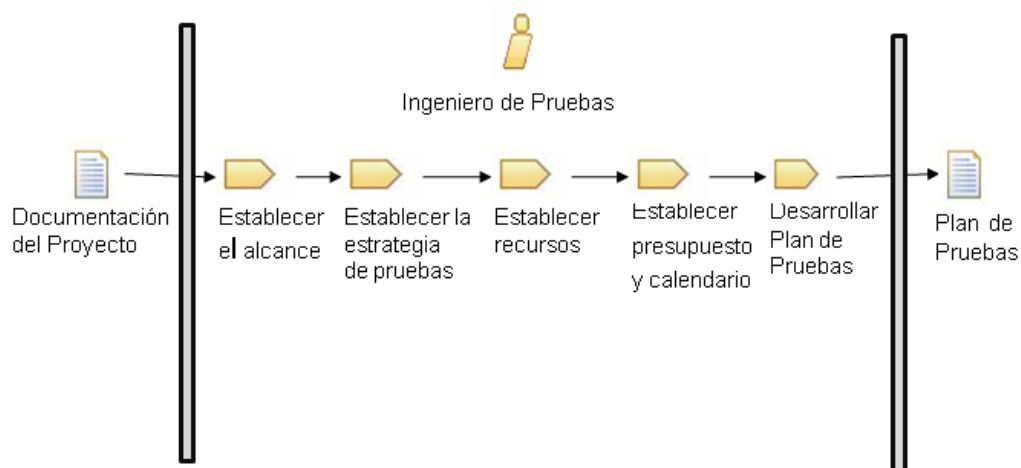
### **Actividad 1: Planificación de las pruebas**

El objetivo de esta actividad es desarrollar el Plan de Pruebas de Aceptación para conseguir que un proceso sea controlable y por tanto las pruebas tengan éxito. La planificación debe comenzar al inicio del proyecto para poder realizar una estimación del tiempo, presupuesto y recursos real, tomando como base la documentación del proyecto.

Para realizar esta actividad se recomienda definir el alcance de las pruebas así como la estrategia de pruebas que se vaya a aplicar; para lo cual habrá que analizar los riesgos del proyecto, priorizar las características que se vayan a probar y conocer la cobertura de la que se puede disponer teniendo en cuenta las restricciones de tiempo o económicas. Además, se tendrá que identificar los recursos tanto personales como materiales para poder asignar las distintas responsabilidades a los miembros que vayan a realizar las actividades de pruebas.

En esta actividad es necesario determinar el presupuesto y el calendario a seguir; los cuales estarán basados en la estimación del tiempo y esfuerzo necesario para desarrollar las actividades de prueba.

Por último, se realiza el Plan de Pruebas de Aceptación usando la información anterior y se obtiene el documento de dicho plan además del documento que contiene el programa de formación.



**Figura 21: Actividad 1: Planificación de las pruebas**

## **Actividad 2: Especificación de los casos de prueba**

El objetivo de esta actividad es recopilar toda la información necesaria para realizar una ejecución efectiva de las pruebas. La elaboración del Plan de Pruebas no es suficiente por no tener en cuenta detalles importantes en la ejecución de las pruebas como puede ser cuales son las entradas requeridas, las salidas esperadas o el procedimiento que se debe seguir en cada una de ellas.

La especificación de los casos de prueba se lleva a cabo después de que los requisitos son aprobados y no cuando el software ha sido desarrollado. De esta forma se definen casos de prueba orientados a los requisitos y se debe tener en cuenta que si se produce alguna modificación en los requisitos o se añade alguno nuevo, será necesaria la revisión de la especificación de los casos de prueba.

Para realizar esta actividad, se parte de la documentación del proyecto así como del Plan de pruebas que ha sido obtenido en la actividad anterior. Se comienza por especificar el diseño de pruebas haciendo uso de las características a probar y de las pruebas asociadas a dichas características. Esta especificación del diseño dará lugar a una lista con los casos de prueba definidos.

Por último, se obtiene una lista con la especificación de los procedimientos de prueba y en ella se especifican los pasos que se deben realizar durante la ejecución de los casos de prueba.



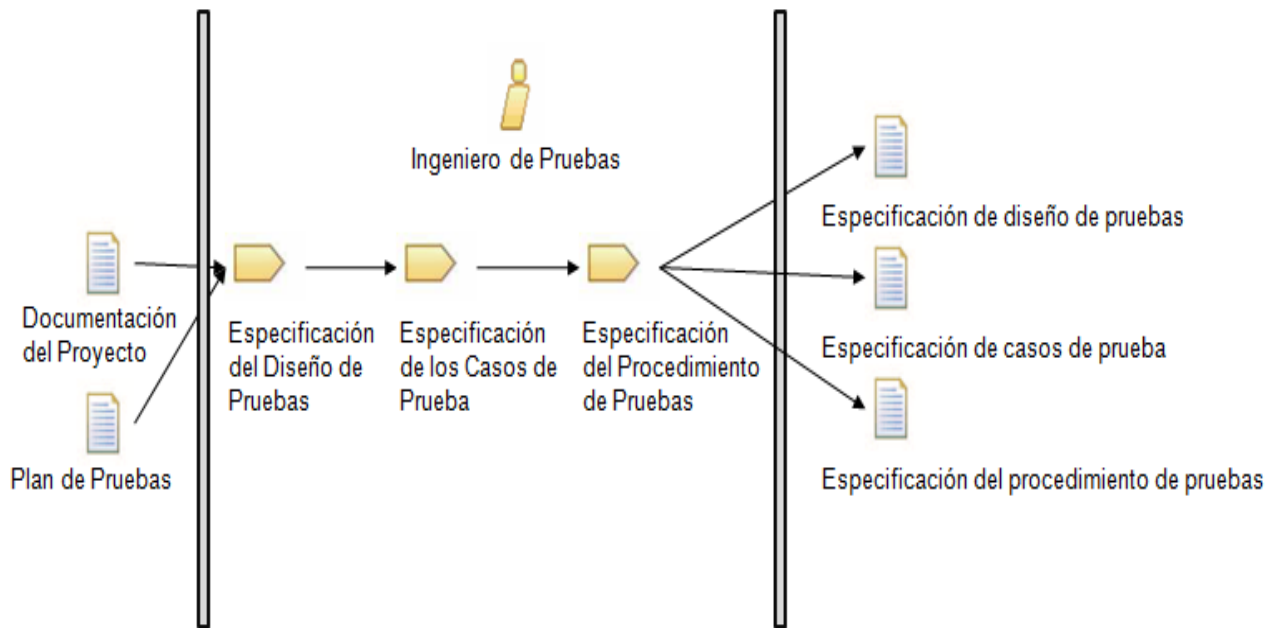


Figura 22: Actividad 2: Especificación de los casos de prueba

### Actividad 3: Preparar el entorno de pruebas de aceptación

El objetivo de esta actividad es llevar a cabo la preparación del entorno necesario para ejecutar el conjunto de pruebas de sistema que ha sido especificado, tomando como base las especificaciones de las pruebas de aceptación y el Plan de Proyecto de Pruebas. El propósito de esta actividad es evaluar que el producto funciona correctamente.

Para realizar esta actividad, se recomienda preparar el hardware, software, librerías y datos para poder llevar a cabo la realización de las pruebas así como generar los datos que se consideren necesarios en la ejecución de los casos de prueba. Cada caso de prueba tendrá un código asociado a que debe ser generado y ejecutado para que cuando se lleve a cabo la ejecución esté disponible. Por último, se debe crear las bases de datos o ficheros que se encarguen de soportar el almacenamiento y recuperación de los datos manejados por el sistema.

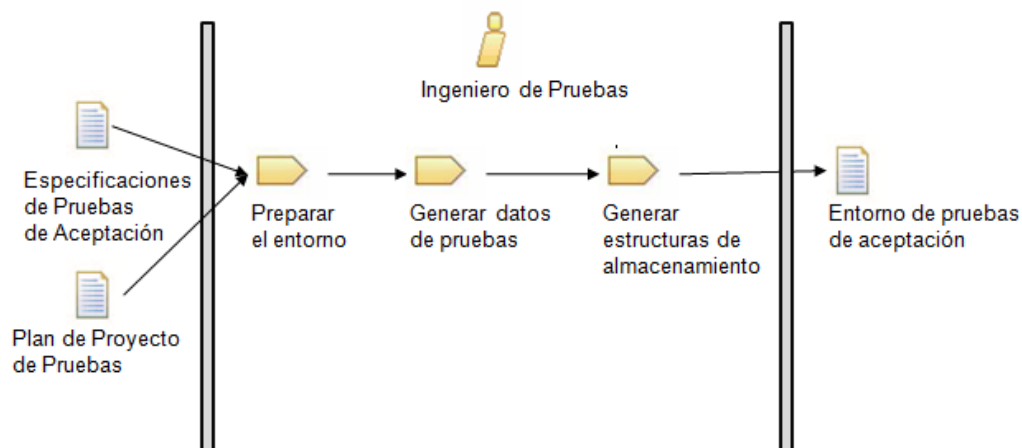


Figura 23: Actividad 3: Preparar el entorno de pruebas de aceptación

#### Actividad 4: Ejecutar los casos de prueba

El objetivo de esta actividad es realizar la ejecución de los casos de prueba generados, tomando como base las especificaciones de las pruebas de Aceptación, el Plan de Proyecto de Pruebas y el entorno de pruebas de Aceptación que ha sido generado.

Para realizar esta actividad, se recomienda determinar mediante el uso de la ficha de pruebas de sistema qué casos de prueba son los que van a ser automatizados para después utilizar los script generados y respetar la secuencia en la que se deben ejecutar las pruebas.

Finalmente, se obtiene un documento que contiene los resultados obtenidos tras la ejecución de las pruebas además de un Test Log con los detalles relevantes de la ejecución.

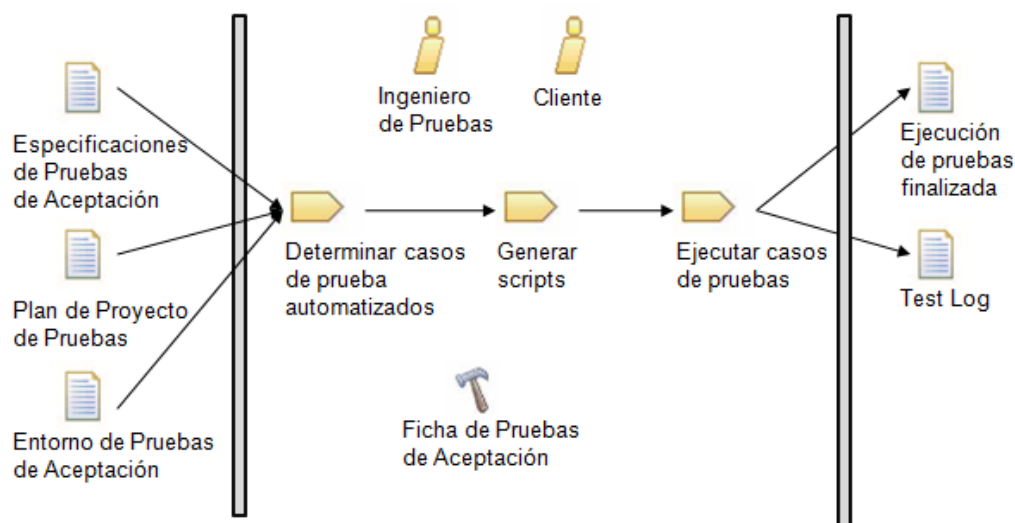


Figura 24: Actividad 4: Ejecutar los casos de prueba

#### Actividad 5: Evaluar pruebas de aceptación

El objetivo de esta actividad es analizar el resultado obtenido de la ejecución de las pruebas de aceptación y poder evaluar en qué medida han sido cumplidos los requisitos. Para ello, se comparará el resultado obtenido durante la ejecución con el resultado que se espera; tomando como base la Ficha de Pruebas de Aceptación y el Plan de Proyecto de Pruebas.

Para comenzar esta actividad, se recomienda comenzar por comprobar si se ha producido algún fallo en la ejecución y en caso de ser afirmativo el defecto se deberá encontrar para posteriormente depurarlo y volver a repetir las pruebas. Una vez las pruebas han sido repetidas se comprueba que la ejecución ha sido satisfactorias. En caso contrario, se debe determinar cuáles son las condiciones anormales que se están dando que hacen que la ejecución no sea adecuada.

Por último, se evalúan los resultados obtenidos y si la evaluación es positiva se entrega al cliente un informe de resultados. En caso contrario, se obtiene un informe de incidentes que será entregado al Cliente y al Jefe de Proyecto, quién se encargará de tomar las acciones que estime necesarias para su resolución.

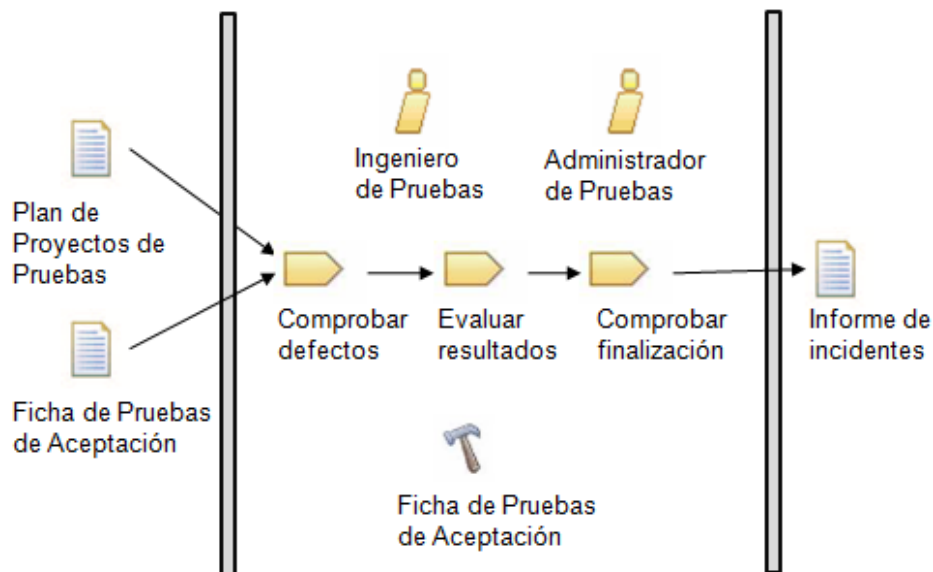


Figura 25: Actividad 5: Evaluar pruebas de aceptación

### ➤ Proceso: Pruebas de Sistema

El propósito de este proceso es llevar a cabo el conjunto de pruebas de sistema que ha sido especificado para evaluar que un producto funciona correctamente.

La figura 26 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo.

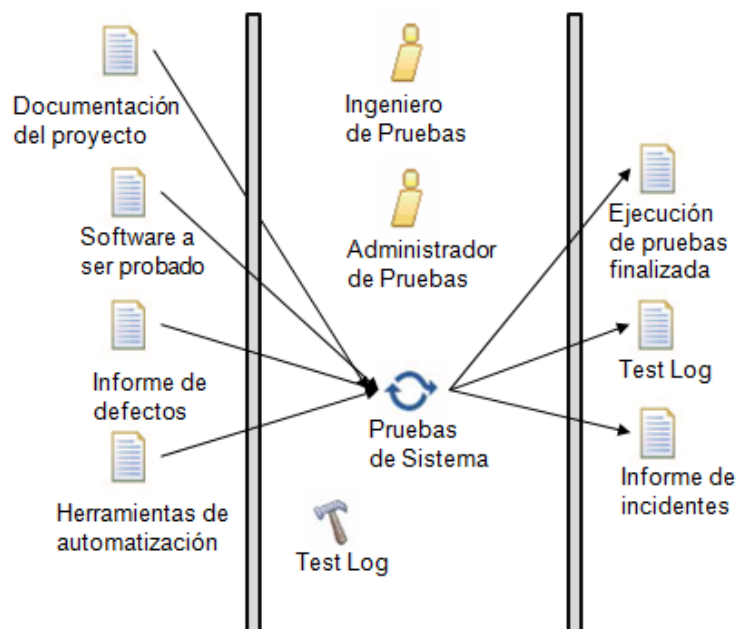


Figura 26: Proceso Pruebas de Sistema

Además de las pruebas funcionales existen diferentes tipos de pruebas de sistema como son:

- **Pruebas de rendimiento:** Validan que se cumple con los requisitos de rendimiento (tiempos de respuesta, tasas de transacción, etc.).
- **Pruebas de seguridad:** Validan que existen las funciones o medidas de seguridad adecuadas para asegurar la integridad y confidencialidad de los datos.
- **Pruebas de volumen:** Someten a la aplicación a grandes volúmenes de datos para determinar si puede manejarlos.
- **Pruebas de carga (stress):** Investigan el comportamiento del sistema bajo condiciones de sobrecarga.
- **Pruebas de compatibilidad:** Comprueban la compatibilidad de la aplicación con otras aplicaciones o sistemas.
- **Pruebas de conversión:** Verifican la conversión de los datos existentes y los cargados en una nueva base de datos.
- **Pruebas de usabilidad:** Determinan cómo de bien utiliza y entiende el usuario la aplicación.
- **Pruebas de recuperación:** Verifican que el sistema puede recuperarse correctamente de un fallo de software o hardware.
- **Pruebas de instalación:** Verifican si el sistema puede instalarse correctamente.
- **Pruebas de regresión:** Validan que el sistema continua funcionando del mismo modo a como lo hacía antes de subsanar defectos encontrados

En la figura 27, se muestran las distintas actividades que definen el proceso.



Figura 27: Actividades del proceso de Pruebas de Sistema

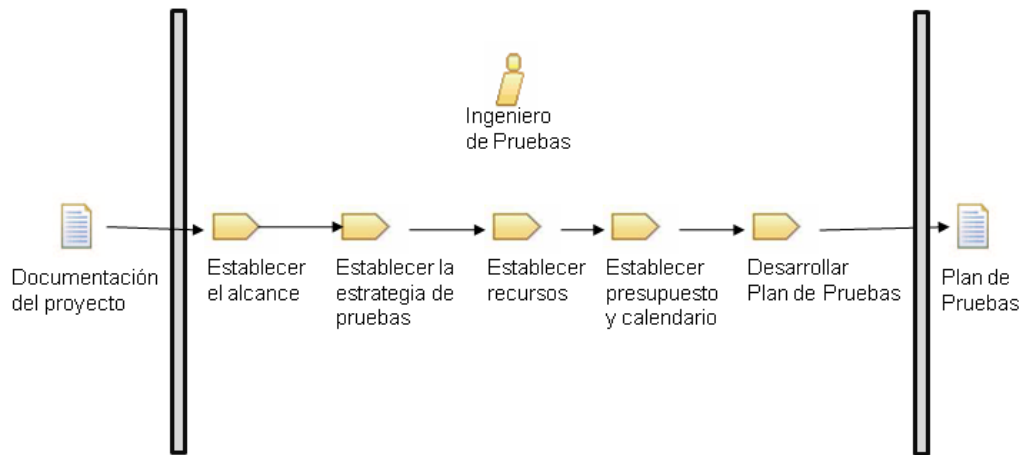
### Actividad 1. Planificación de las pruebas

El objetivo de esta actividad es desarrollar el Plan de Pruebas de Sistema para conseguir que un proceso sea controlable y por tanto las pruebas tengan éxito. La planificación de las pruebas debe comenzar al inicio del proyecto para poder realizar una estimación del tiempo, presupuesto y recursos real, tomando como base la documentación del proyecto.

Para realizar esta actividad se recomienda definir el alcance de las pruebas de sistema así como la estrategia de pruebas que se vaya a aplicar; para lo cual habrá que analizar los riesgos del proyecto, priorizar las características que se vayan a probar y conocer la cobertura de la que se puede disponer teniendo en cuenta las restricciones de tiempo o económicas. Además, se tendrá que identificar los recursos tanto personales como materiales para poder asignar las distintas responsabilidades a los miembros que vayan a realizar las actividades de pruebas.

En esta actividad es necesario determinar el presupuesto y el calendario a seguir, los cuales estarán basados en la estimación del tiempo y esfuerzo necesario para desarrollar las actividades de prueba.

Por último, se desarrolla el Plan de Pruebas de Sistema usando la información anterior.



**Figura 28: Actividad 1: Planificación de las pruebas**

## **Actividad 2: Especificación de casos de prueba**

El objetivo de esta actividad es recopilar toda la información necesaria para realizar una ejecución efectiva de las pruebas. La elaboración del Plan de Pruebas no es suficiente por no tener en cuenta detalles importantes en la ejecución de las pruebas; como puede ser cuales son las entradas requeridas, las salidas esperadas o el procedimiento que se debe seguir en cada una de ellas.

La especificación de los casos de prueba se lleva a cabo después de que los requisitos son aprobados y no cuando el software ha sido desarrollado. De esta forma se definen casos de prueba orientados a los requisitos y se debe tener en cuenta que si se produce alguna modificación en los requisitos o se añade alguno nuevo, será necesaria la revisión de la especificación de los casos de prueba.

Para realizar esta actividad, se parte de la documentación del proyecto así como del Plan de Pruebas que ha sido obtenido en la actividad anterior. Se comienza por especificar el diseño de pruebas haciendo uso de las características a probar y de las pruebas asociadas a dichas características.

Esta especificación del diseño establecerá una lista con los casos de prueba definidos y por último se obtiene la especificación de los procedimientos de prueba, es decir, los pasos que se deben realizar durante la ejecución de los casos de prueba.

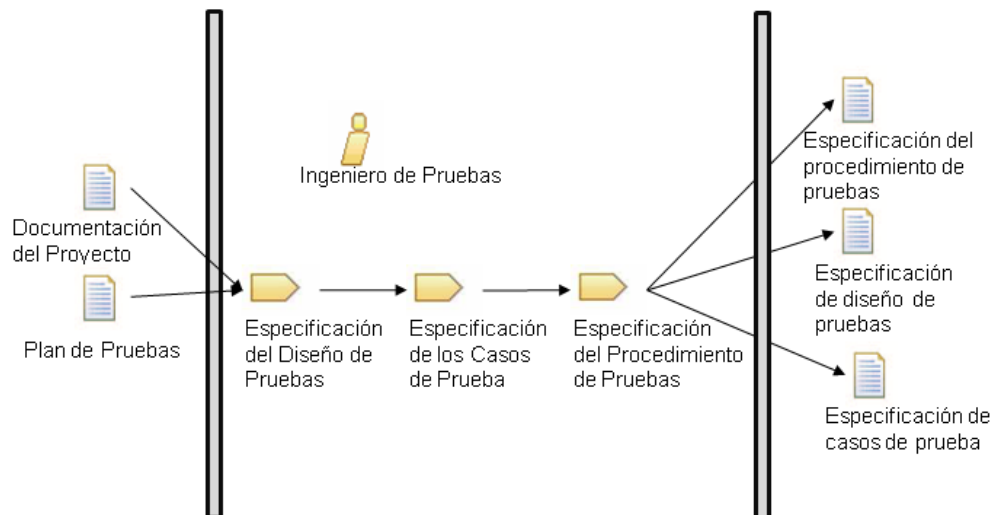


Figura 29: Actividad 2: Especificación de casos de prueba

### Actividad 3: Preparar el entorno de de pruebas del sistema

El objetivo de esta actividad es llevar a cabo la preparación del entorno necesario para ejecutar el conjunto de pruebas de aceptación que ha sido especificado tomando como base las especificaciones de las pruebas de sistema y el Plan de Pruebas. El objetivo de esta actividad es evaluar que el producto funciona correctamente.

Para realizar esta actividad, se recomienda preparar el hardware, software, librerías y datos necesarios para poder llevar a cabo la realización de las pruebas así como generar los datos de pruebas necesarios en la ejecución de los casos de prueba. Cada caso de prueba tendrá un código asociado a que debe ser generado y ejecutado para que cuando se lleve a cabo la ejecución esté disponible. Por último, se debe crear las bases de datos o ficheros que se encarguen de soportar el almacenamiento y recuperación de los datos manejados por el sistema.

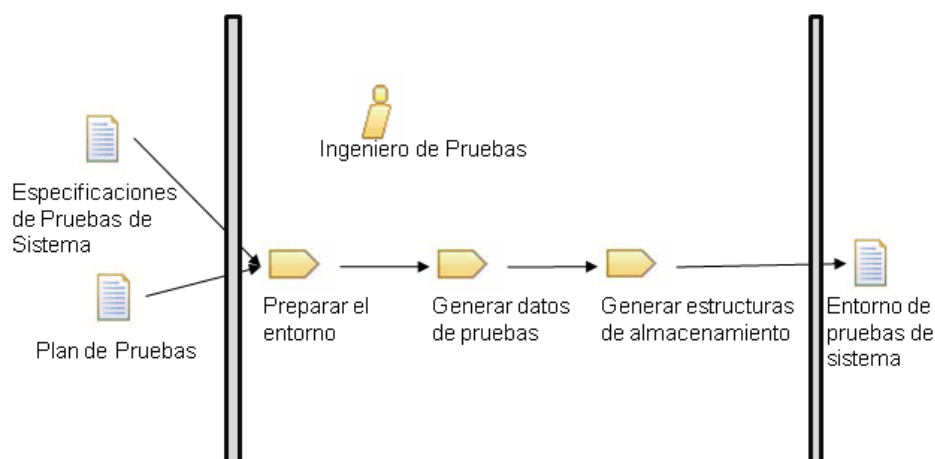


Figura 30: Actividad 3: Preparar el entorno de de pruebas del sistema

#### Actividad 4: Ejecutar los casos de prueba

El objetivo de esta actividad es realizar la ejecución de los casos de prueba generados, tomando como base las Especificaciones de las Pruebas de Sistema, el Plan de Proyecto de Pruebas y el Entorno de Pruebas de Sistema que ha sido generado.

Para realizar esta actividad, se recomienda determinar mediante el uso de la ficha de pruebas de sistema qué casos de prueba son los que van a ser automatizados para después utilizar los script generados y respetar la secuencia en la que se deben ejecutar las pruebas.

Finalmente, se obtiene un documento que contiene los resultados obtenidos tras la ejecución de las pruebas además de un Test Log con los detalles relevantes de la ejecución.

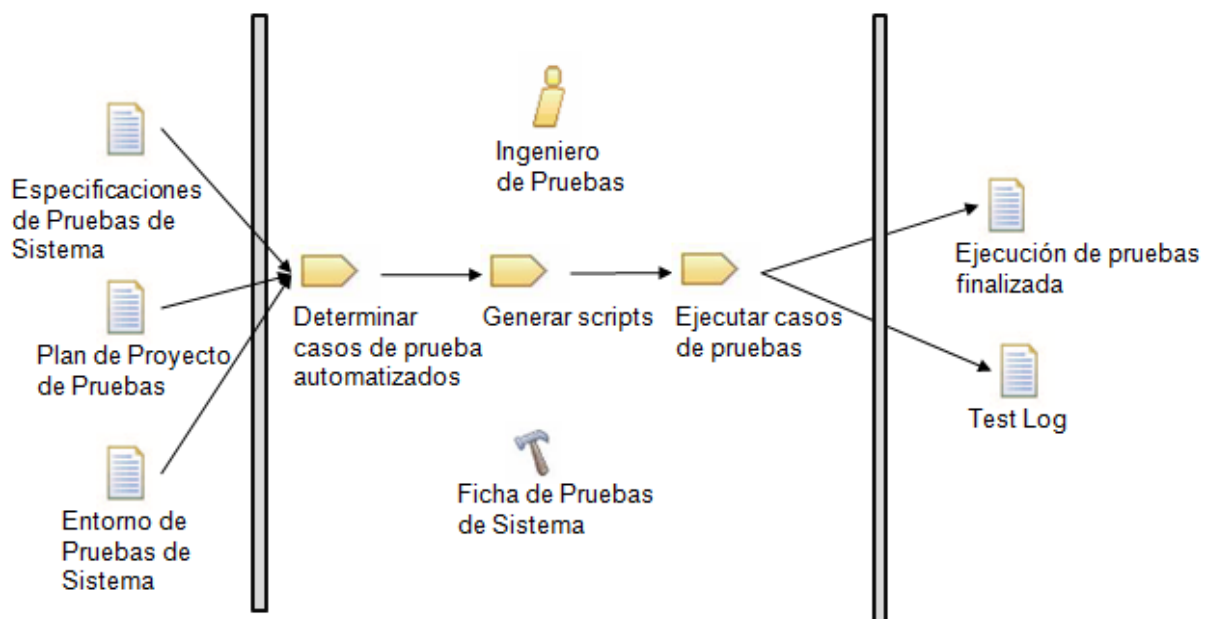


Figura 31: Actividad 4: Ejecutar los casos de prueba

#### Actividad 5: Evaluar pruebas de sistema

El objetivo de esta actividad es analizar el resultado obtenido de la ejecución de las pruebas de sistema y poder evaluar en qué medida han sido cumplidos los requisitos. Para ello, se comparará el resultado obtenido durante la ejecución con el resultado que se espera, tomando como base la Ficha de Pruebas de Sistema y el Plan de Proyecto de Pruebas.

Para comenzar esta actividad, se recomienda comenzar por comprobar si se ha producido algún fallo en la ejecución y en caso de ser afirmativo el defecto se deberá encontrar para posteriormente depurarlo y volver a repetir las pruebas. Una vez las pruebas han sido repetidas se comprueba que la ejecución ha sido satisfactorias. Por el contrario, se debe determinar cuáles son las condiciones anormales que se están dando que hacen que la ejecución no sea adecuada.

Por último, se evalúan los resultados obtenidos y si la evaluación es positiva se entrega al cliente un informe de resultados. En caso contrario, se obtiene un informe de incidentes

que será entregado al cliente y al Jefe de Proyecto, quién se encargará de tomar las acciones que estime necesarias para su resolución.

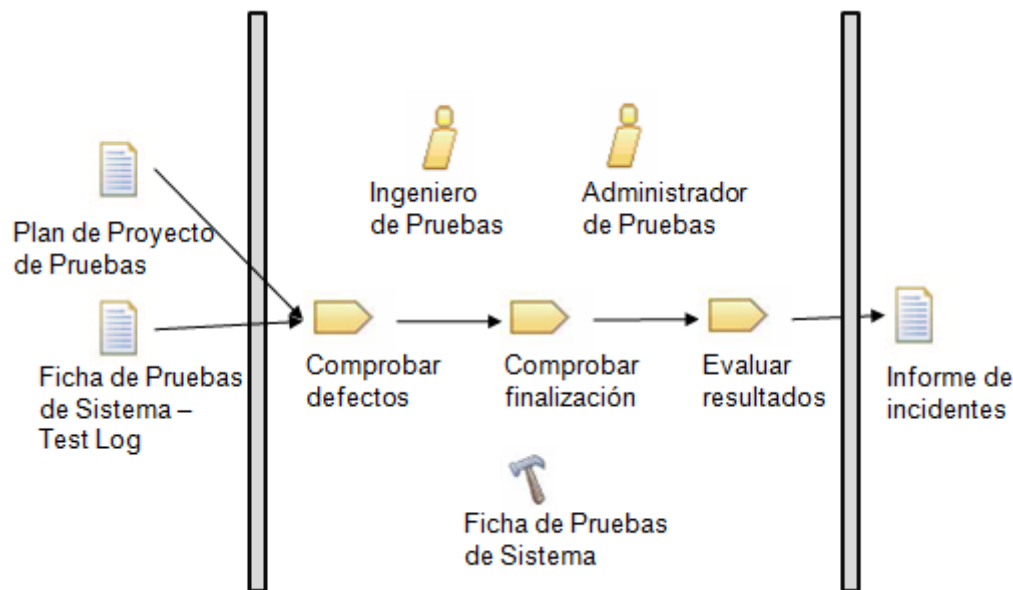


Figura 32: Actividad 5: Evaluar pruebas de sistema

### ➤ Proceso: Verificación Código Fuente

El objetivo de este proceso es asegurar que el código fuente no contiene ningún defecto de corrección como puede ser bucles infinitos, variable definida de un tipo pero utilizada de otro, contador que se sale de las dimensiones de un array ni defectos de validez que hacen que el código no se corresponda con el diseño establecido.

La figura 33 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo.

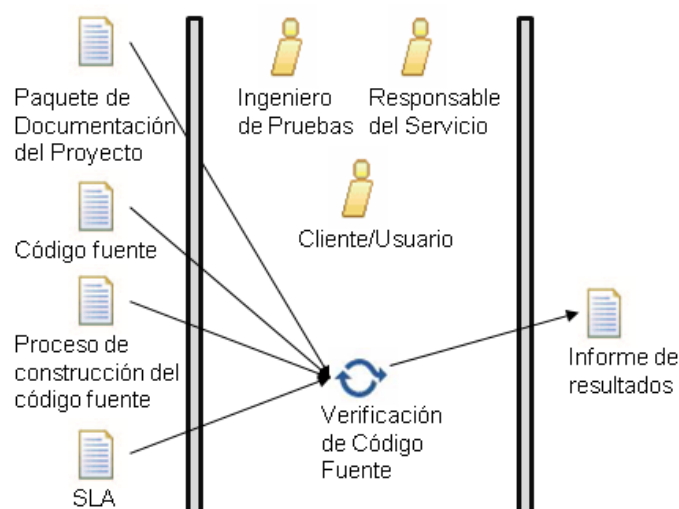


Figura 33: Proceso Verificación Código Fuente

Existen un conjunto de características de calidad que permite especificar y evaluar la calidad de un producto software desde diferentes perspectivas. Estas características son:



**Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad.** Al realizar un análisis exhaustivo del código fuente se puede determinar el grado de cumplimiento de cada una de estas características.

En la figura 34, se muestran las distintas actividades que definen el proceso.

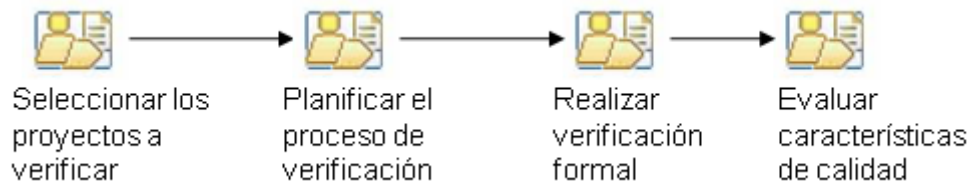


Figura 34: Actividades del proceso Verificación Código Fuente

### Actividad 1: Seleccionar los proyectos a verificar

El objetivo de esta actividad es seleccionar el conjunto de proyectos sobre los que se desea aplicar el proceso de verificación. Para ello, se tendrá en cuenta el tamaño, naturaleza y número de personas que integran el equipo de trabajo tomando como base la Lista de proyectos y los Criterios de selección.

Para realizar esta actividad, se recomienda comenzar por identificar que proyectos se van a usar en la verificación del código fuente. Para ello, se siguen los criterios de selección previamente definidos y el resultado obtenido es la lista con los proyectos que han sido seleccionados. Una vez identificados los proyectos, se establece el orden de verificación del código fuente de los proyectos seleccionados previamente.

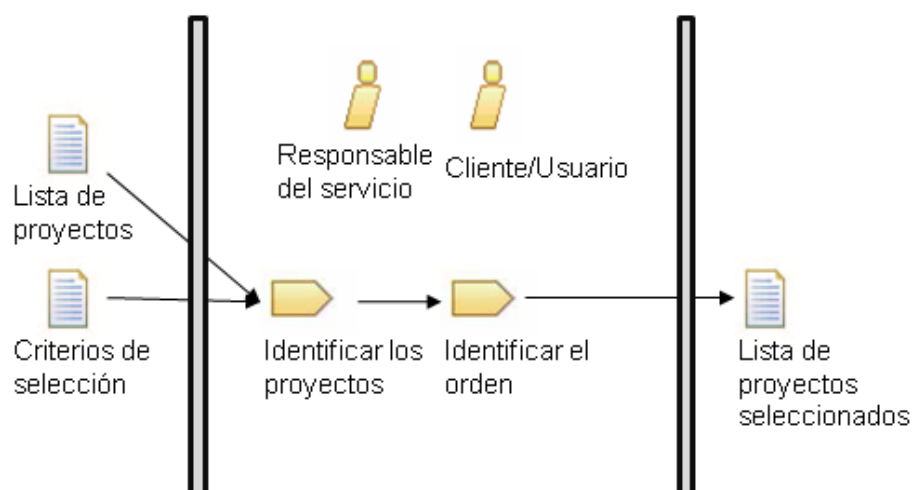


Figura 35: Actividad 1: Seleccionar los proyectos a verificar

## Actividad 2: Planificar el proceso de verificación

El objetivo principal de esta actividad es establecer la planificación a seguir durante el desarrollo de la verificación de código.

Para realizar esta actividad, se recomienda comenzar por identificar las actividades a desempeñar en el proceso de verificación para establecer el orden de precedencia entre ellas y asignar a cada actividad un responsable que juntos formarán el equipo de trabajo donde cada miembro tiene su rol asignado.

Por último, se debe definir un calendario para realizar de manera ordenada las actividades que han sido planificadas además de identificar todos los recursos necesarios para llevar a cabo cada una de las actividades de la planificación del proceso de verificación del código fuente.

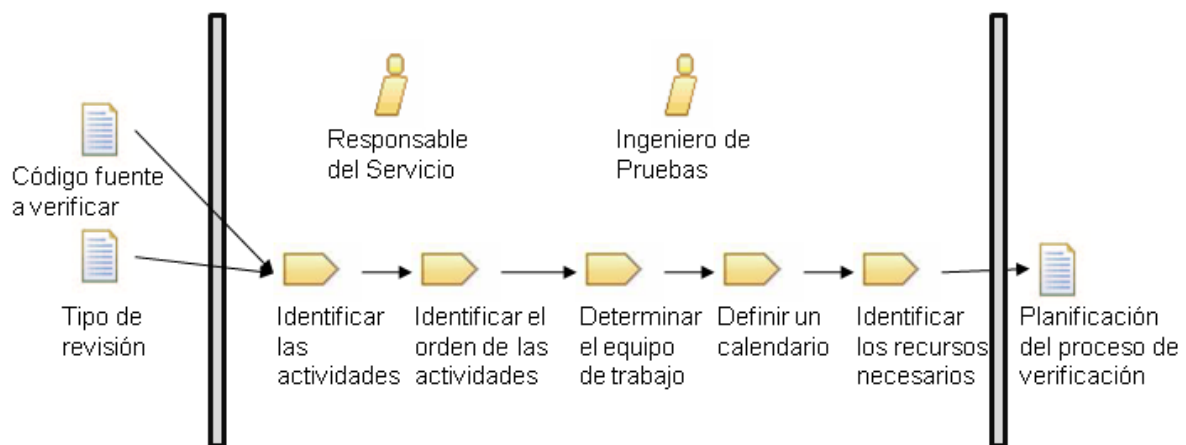


Figura 36: Actividad 2: Planificar el proceso de verificación

## Actividad 3: Realizar verificación formal

El objetivo de esta actividad es realizar la verificación del código fuente de un proyecto a través de una verificación formal, tomando como base el código fuente a verificar y el SLA (Acuerdo de Nivel del Servicio). Se empleará análisis estático para verificar el código fuente debido a que este análisis reconoce y arregla defectos sin necesidad de ejecutar el código. Para llevar a cabo este análisis se deben llevar a cabo las siguientes etapas:

- **Análisis del flujo de control:** Comprueba los bucles con múltiples entradas y salidas.
- **Análisis del uso de los datos:** Detecta variables no inicializadas, variables que se declaran pero nunca se usan.
- **Análisis de interfaz:** Comprueba la consistencia de una rutina, las declaraciones de un procedimiento y su uso.
- **Análisis de flujo de información:** Identifica las dependencias de las variables de salida.
- **Análisis de caminos:** Identifica los caminos del programa y arregla las sentencias ejecutadas en el camino.

Para realizar esta actividad, se recomienda comenzar por familiarizar al equipo de trabajo con el código que va a ser verificado para posteriormente cada inspector de forma individual revise el código usando una checklist que contiene toda la información que se debe verificar del código para después elaborar la lista de defectos encontrados.

Una vez revisado el código, se celebra una reunión en la que se ponen en común los defectos detectados para posteriormente corregirlos y obtener como resultado la lista de defectos subsanados.

Por último, se comprueba que la revisión ha sido realizada y se elabora un informe que contiene los principales resultados de la verificación del código.



Figura 37: Actividad 3: Realizar verificación formal

#### Actividad 4: Evaluar características de calidad

El objetivo de esta actividad es determinar el grado de cumplimiento de las seis características de calidad de un software. Estas características son: **funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad**. Cada una de estas características tiene unos atributos que se miden mediante métricas internas que miden las propiedades intrínsecas y externas que se ocupan del comportamiento del sistema y por las cuales se comprueba que el software proporciona calidad. Para ello se tomará como base el código fuente que se va a verificar y el SLA (Acuerdo de Nivel del Servicio).

Para realizar esta actividad, se recomienda comenzar por identificar las métricas internas y externas que se van a usar para evaluar los atributos de calidad para después saber que procedimiento se va a seguir en el análisis de las características que se van a evaluar. Una vez definido el procedimiento, se llevan a cabo las actividades necesarias para evaluar las características y conocer el grado de satisfacción de cada una de ellas.

Por último, se elabora un informe que contiene todos los resultados obtenidos tras la evaluación.

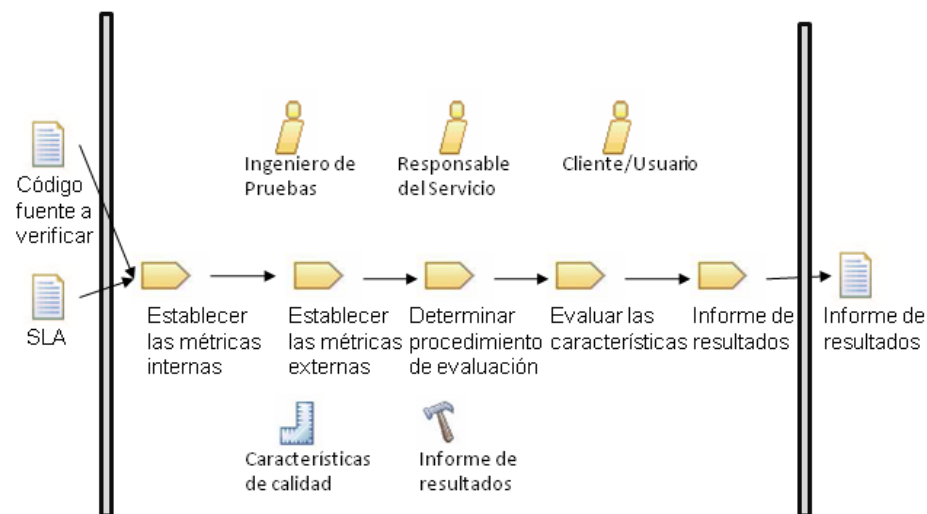


Figura 38: Actividad 4: Evaluar características de calidad

### ➤ Proceso: Verificación de Productos de Trabajo

El objetivo de este proceso es asegurar que los productos que intervienen en el proceso de desarrollo de software son correctos, consistentes y completos, es decir satisfacen las necesidades del cliente.

La figura 39 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo

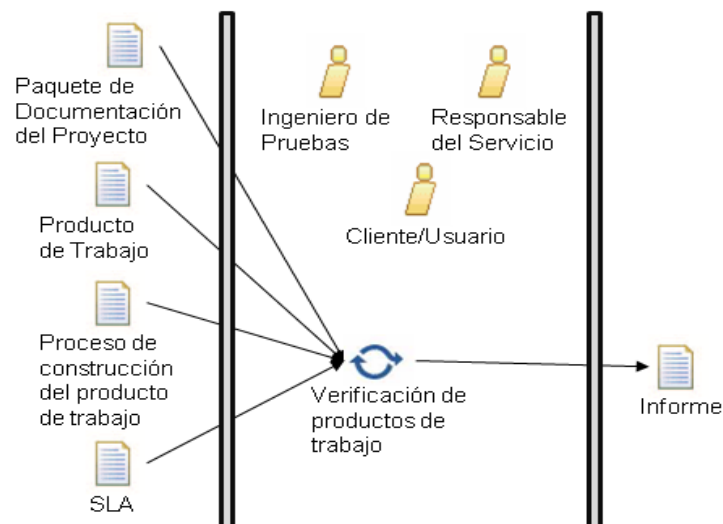
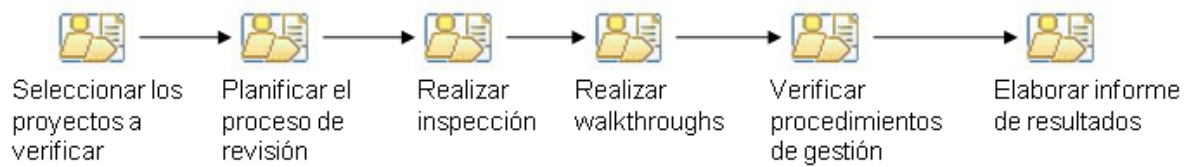


Figura 39: Proceso Verificación de Productos de Trabajo

El principal motivo de llevar a cabo una verificación de productos intermedios es evitar defectos en el producto final e incrementar la calidad del producto al eliminar los defectos así como reducir el re-trabajo y esfuerzo en fases de desarrollo posteriores.

En la figura 40, se muestran las distintas actividades que definen el proceso.



### Actividad 1: Seleccionar los proyectos a verificar

El objetivo de esta actividad es seleccionar aquellos productos sobre los cuales se va a realizar el proceso de verificación tomando como base la lista de proyectos disponibles así como la lista de criterios a emplear en la selección.

Para realizar esta actividad se identifican los proyectos a utilizar según criterios como el tamaño y el número de personas que integran el equipo de trabajo y de la cual se obtiene una lista para posteriormente establecer el orden en el cual van a ser revisados los productos de trabajo de los proyectos que han sido seleccionados previamente.

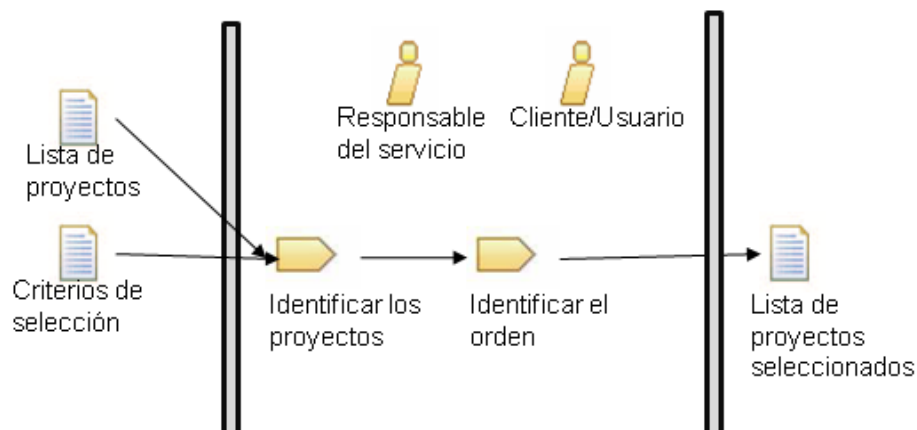
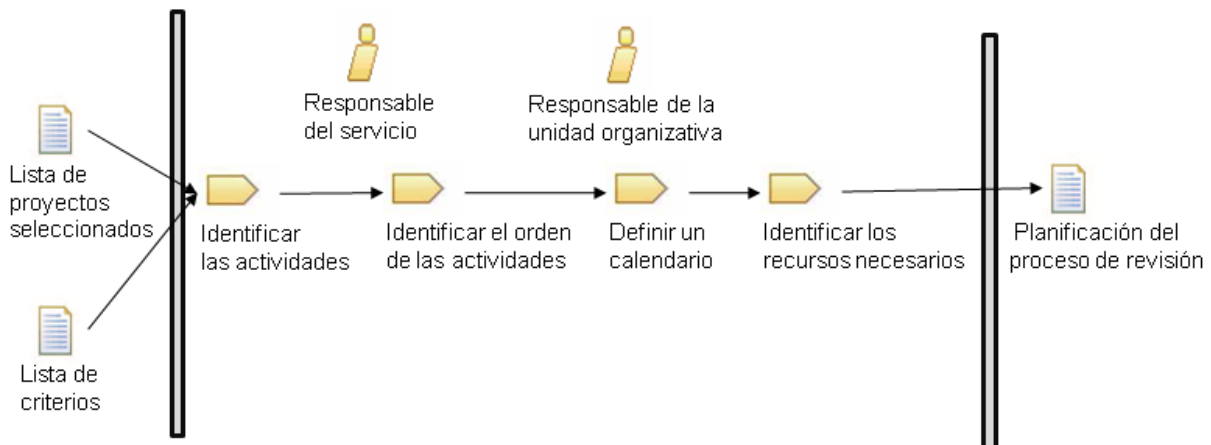


Figura 41: Actividad 1: Seleccionar los proyectos a verificar

### Actividad 2: Planificar el proceso de revisión

El objetivo de esta actividad es planificar el proceso de revisión tomando como base la lista de proyectos seleccionados así como la lista de criterios.

Para realizar esta actividad se determinarán las actividades que se van a realizar y quién las realiza, el período y lugar de ejecución, los recursos necesarios, número de versiones así como el alcance de las revisiones y la definición de un calendario a seguir durante la verificación.



**Figura 42: Actividad 2: Planificar el proceso de revisión**

### **Actividad 3: Realizar Inspección**

El objetivo de esta actividad es inspeccionar los productos o entregables de un proyecto cuyo propósito principal es detectar defectos antes de llegar a la fase de validación del sistema.

Para realizar esta actividad se identifican los defectos durante las pruebas dinámicas y esto tienen asociado un mayor coste tanto monetario como en tiempo, ya que además de modificar el software para subsanar el fallo será necesario modificar toda la documentación de la que depende; mientras que si identificamos un defecto durante el desarrollo de las pruebas estáticas únicamente habrá que modificar la documentación afectada.

Las principales ventajas de realizar una inspección son que es una técnica sistemática, controlada y poco estresante, que si se gestiona adecuadamente puede ser un foro en el cual los desarrolladores no necesiten protegerse.

Los inspectores realizarán una revisión de forma individual y posteriormente, una en equipo con el objetivo de alcanzar un consenso en los errores encontrados. Estos defectos serán comunicados a quién corresponda con el objetivo de que sean subsanados antes de llegar a la ejecución de las pruebas dinámicas o validación del producto. Una vez que estos cambios sean realizados, será necesario volver a ejecutar una revisión para comprobar que todo se ha desarrollado correctamente.

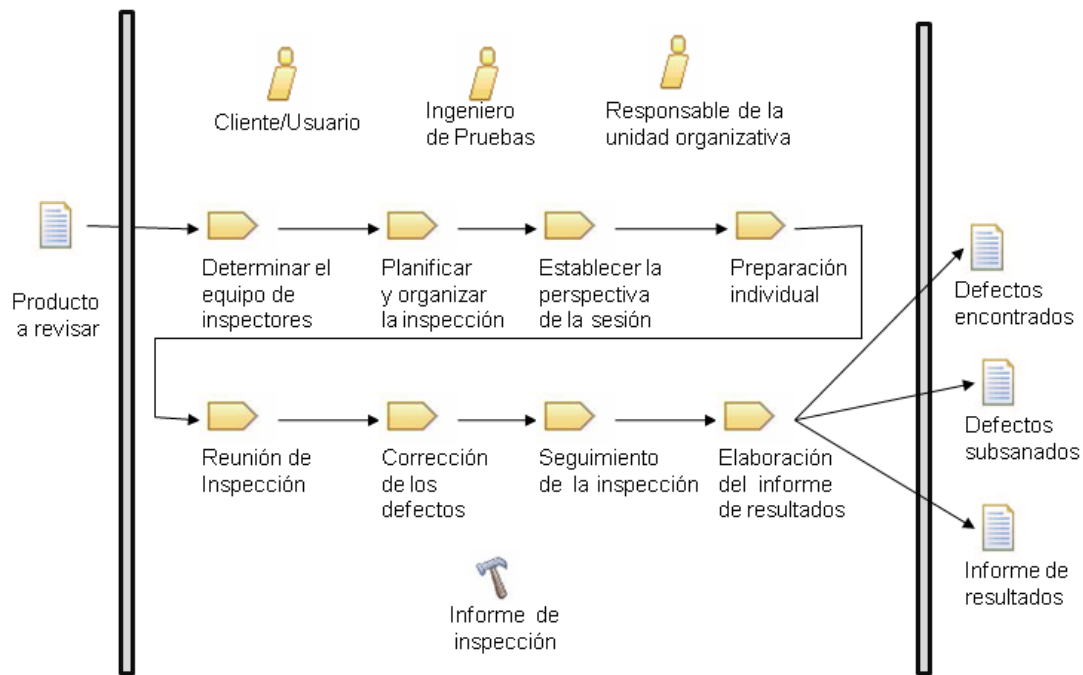


Figura 43: Actividad 3: Realizar Inspección

#### Actividad 4: Realizar Walkthroughs

El objetivo principal de esta actividad es llevar a cabo una revisión informal o walkthrough para obtener nuevas soluciones a un problema determinado. El objetivo principal del walkthrough es crear una situación en la cual un equipo de individuos cualificados pueda ayudar al equipo del proyecto a desarrollar soluciones tomando como base el producto a revisar.

Para realizar esta actividad, es necesario: determinar el número de participantes, conocer las responsabilidades del equipo de walkthrough y mantener la confidencialidad de la información. Una vez las reglas han sido definidas, se eligen a los miembros que forman el equipo de walkthrough así como sus responsabilidades y también los compromisos de tiempo y fecha del walkthrough. El equipo de proyecto presenta el producto al equipo de trabajo para conocerlo y que estos puedan ofrecer sus recomendaciones y por tanto generar nuevas ideas o soluciones.

Por último se genera un informe final que contiene todo los hallazgos obtenidos en la presentación del producto.



Figura 44: Actividad 4: Realizar Walkthroughs

### Actividad 5: Verificar procedimientos de gestión

Esta actividad tiene como objetivo revisar las actividades y los procedimientos que han sido realizados para poder asegurar que el proceso implantado se está realizando correctamente. Para ello se toma como base la Definición del proceso y el Producto de Trabajo que ha sido obtenido en el desarrollo del proceso.

Para realizar esta actividad se comienza por identificar las actividades que forman el producto de trabajo para posteriormente formar el equipo de trabajo. Una vez ha sido definido el equipo de trabajo, se recogen y se analizan las evidencias identificadas en esas actividades para a continuación poder revisar si dichas actividades son correctas y consistentes.

Por último, se genera un informe con los hallazgos obtenidos en el análisis y verificación de las actividades del proceso, identificando tanto las fortalezas como las debilidades encontradas.

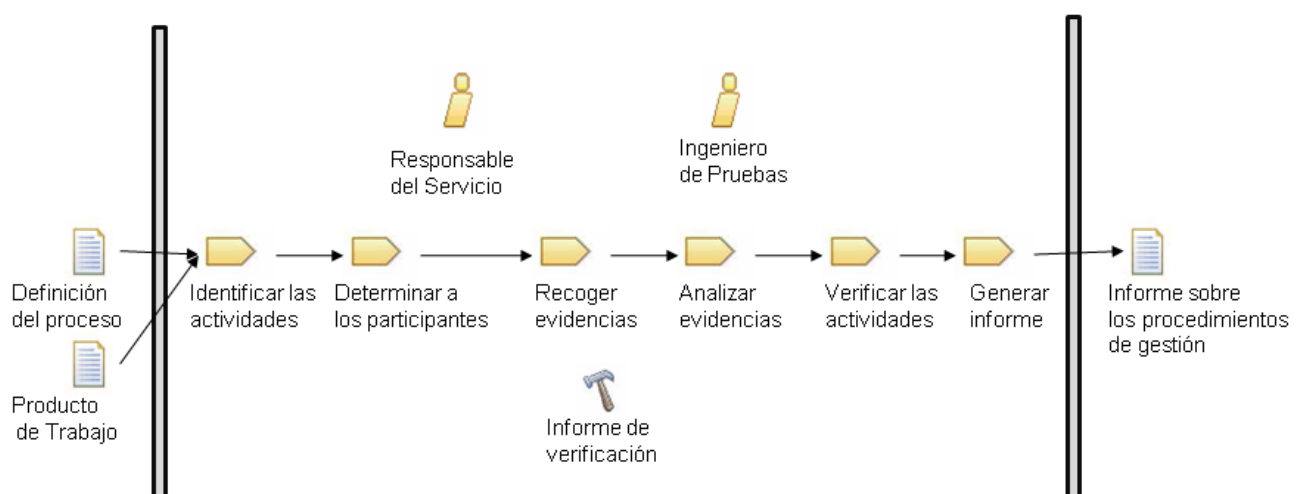


Figura 45: Actividad 5: Verificar procedimientos de gestión



## Actividad 6: Elaborar informe de resultados

Esta actividad tiene como objetivo elaborar el informe final que recoge los resultados de la realización del proceso de verificación. Para ello, se toma como base el Informe de resultados sobre los procedimientos de gestión y el Informe de resultados sobre el Producto de trabajo.

Para realizar esta actividad, se analizan los resultados obtenidos en la verificación del producto de trabajo para poder conocer los problemas y las causas que los provocan. Una vez se conocen los problemas se definen las acciones correctoras a aplicar y se documentan las lecciones aprendidas para finalmente elaborar un informe con los hallazgos obtenidos tras la realización de las tareas previas.



Figura 46: Actividad 6: Elaborar informe de resultados

## ➤ Proceso: Gestión de la Configuración

El objetivo de este proceso es asegurar la integridad de los productos y servicios que se han desarrollado.

La figura 47 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo.

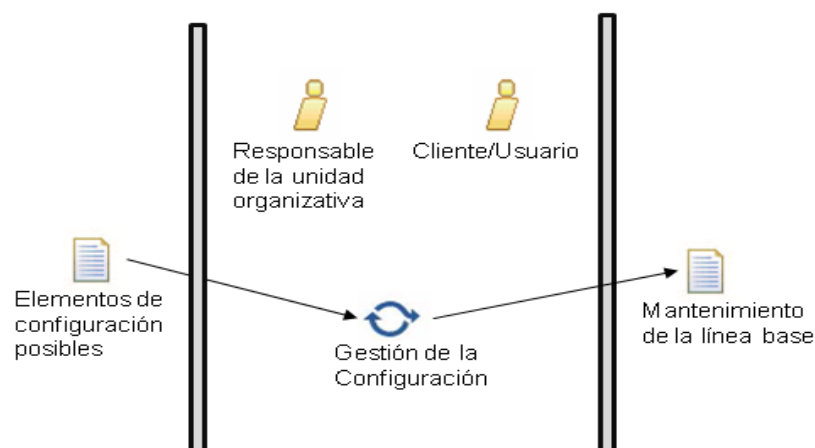


Figura 47: Proceso Gestión de la Configuración

Para llevar a cabo la gestión de la configuración, se identifican los productos que se deben de controlar y se establece las actividades de control que se van a realizar. Toda la información es recogida en un repositorio de datos que se mantendrá actualizado ante posibles cambios y tenga disponible versiones estables de cada uno de los productos que está almacenado. La Gestión de la Configuración implica:

- Identificar la configuración de productos de trabajo que componen la línea base.
- Controlar los cambios de los elementos de configuración.
- Mantener la integridad de la línea base.

En la figura 48, se muestran las distintas actividades que definen el proceso.

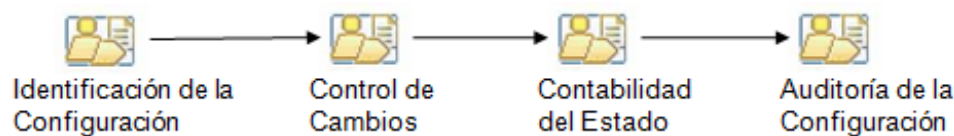


Figura 48: Actividades del proceso Gestión de la Configuración

### **Actividad 1: Identificación de la Configuración**

El objetivo de esta actividad es identificar los elementos que van a formar parte de la configuración y elaborar un Plan de Gestión de Configuración que proporcione el conjunto de actividades de gestión de configuración que tienen que ser realizadas por aquellos que tengan que aprobar, realizar e interactuar con los elementos de configuración. Se registrarán todos aquellos productos de trabajo que son necesarios de controlar en un proyecto ya sean elementos intermedios o finales y se verá su evolución con el paso del tiempo. Se hará un listado con aquellos elementos que van a considerarse como entregables del proyecto.

Para realizar esta actividad, se comienza por identificar los elementos de configuración a los cuales se les asigna un nombre, código de versión, el estado que indica la situación en la que se encuentra dentro del proceso de elaboración y la localización en el sistema de gestión de la configuración de manera que están identificados de manera única. A continuación, se identifica el propósito y alcance del plan de gestión de configuración para posteriormente identificar los recursos necesarios así como las responsabilidades de cada uno.

Por último, se identifican las actividades de gestión de configuración que se van a llevar a cabo, se define el calendario para realizar dichas actividades y se define el entorno tecnológico que da soporte a la gestión de la configuración.

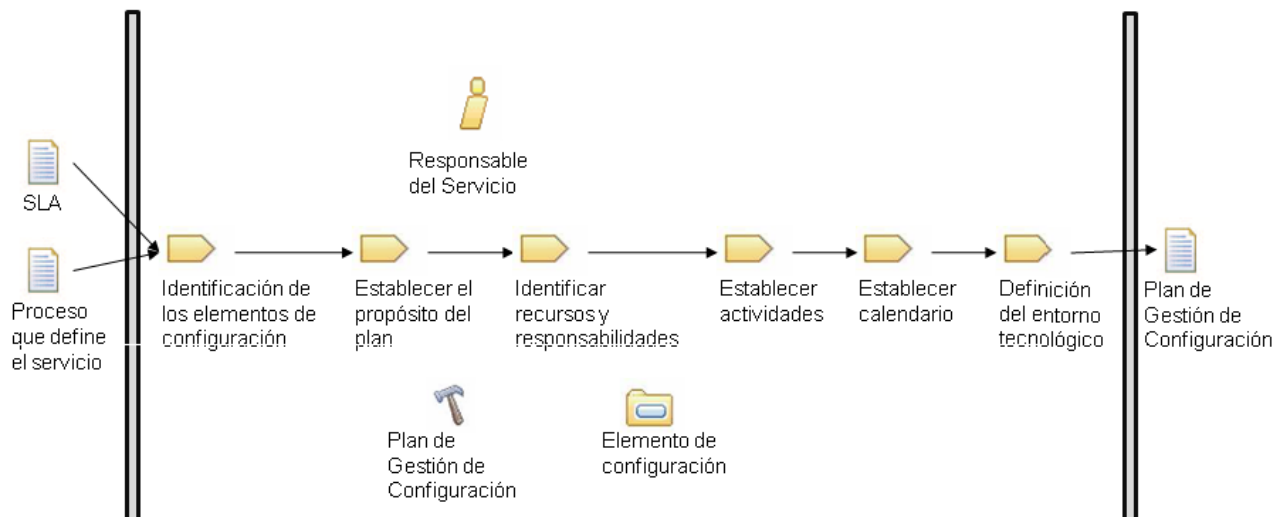


Figura 49: Actividad 1: Identificación de la Configuración

## Actividad 2: Control de Cambios

El objetivo principal de esta actividad es realizar de forma exhaustiva y controlada los posibles cambios sobre los elementos de configuración y las líneas base tomando como base la petición de cambio. Para ello, se define una petición de cambios que especifica las actividades que se han de realizar cada vez que surja un cambio.

El procedimiento de control de cambios comienza por elaborar la petición de cambio que especifique el cambio que se propone realizar y el elemento sobre el cual se realiza. A continuación, se analiza el impacto que tendrá el realizar dicho cambio para posteriormente aceptar o rechazar dicho cambio.

Por último, el estado de la petición de cambio es comunicado a todos los afectados.

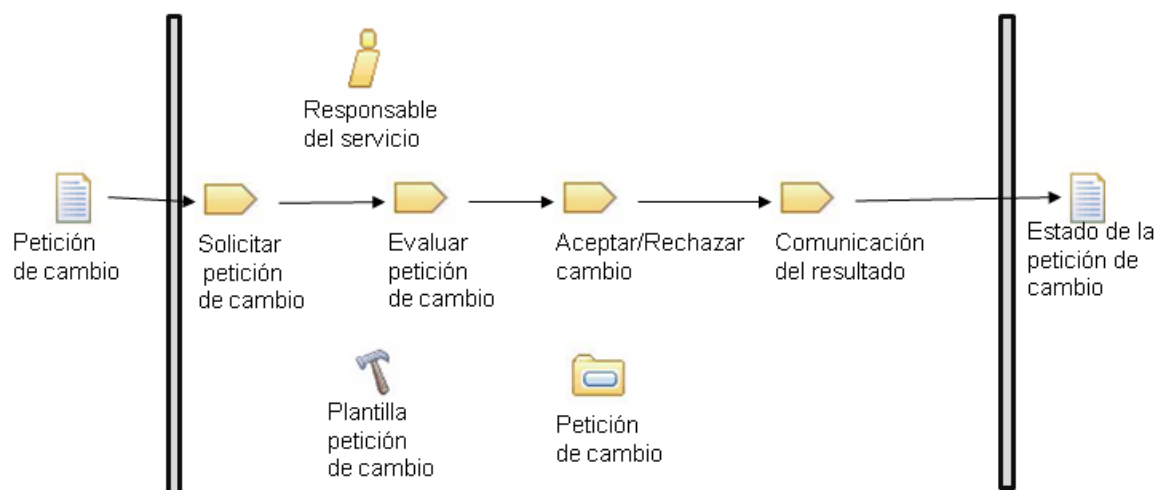
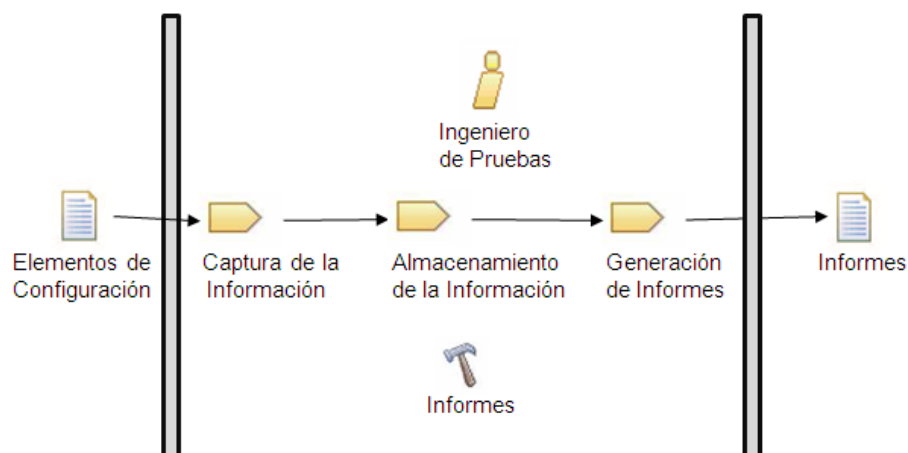


Figura 50: Actividad 2: Control de Cambios

### **Actividad 3: Contabilidad de Estado**

El objetivo principal de esta actividad es mantener a los usuarios, a los gestores y a los desarrolladores informados sobre el estado en el que se encuentra la configuración y cuál es su evolución. Es decir, reciben la información necesaria para saber que ha ocurrido en cada momento y cuando, además de mejorar los problemas de comunicación que puedan surgir entre los miembros del proyecto. Para ello, se toma como base los elementos de configuración, es decir, el conjunto de productos de trabajo sobre los cuales se desea aplicar la gestión de configuración.

La actividad contabilidad de estado comienza por capturar y almacenar toda la información relacionada con el proyecto para posteriormente generar informes que transmitan la información.



**Figura 51:Actividad 3:Contabilidad de Estado**

### **Actividad 4: Auditoría de la Configuración**

El objetivo principal de esta actividad es verificar el resultado de un trabajo para evaluar su grado de conformidad con respecto a posibles especificaciones, estándares, acuerdos contractuales u otros criterios. La auditoria de configuración es la forma de comprobar que efectivamente el producto que se está construyendo es lo que pretende ser. Para ello, se toma como base la línea base de elementos de configuración, es decir, los productos de trabajo que van a ser verificados.

Esta actividad comienza por recoger la información necesaria mediante la realización de un cuestionario. Una vez el cuestionario ha sido rellenado, se documenta el informe de auditoría que contiene las debilidades y fortalezas que han sido encontradas en dicha auditoría.

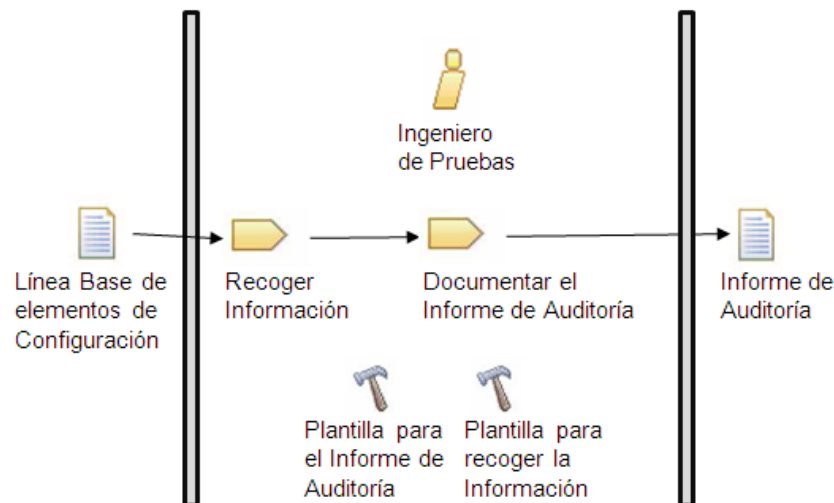


Figura 52: Actividad 4: Auditoría de la Configuración

### ➤ Proceso: Aseguramiento de la Calidad

El objetivo de este proceso es llevar a cabo el seguimiento de las actividades de calidad comprometidas internamente en el servicio. Es decir, ejecutar aquellas acciones que hacen que un proceso cumpla con unos determinados requisitos de calidad.

La figura 53 muestra el conjunto de productos de entrada y salida del proceso, así como los diferentes roles que intervienen en el mismo

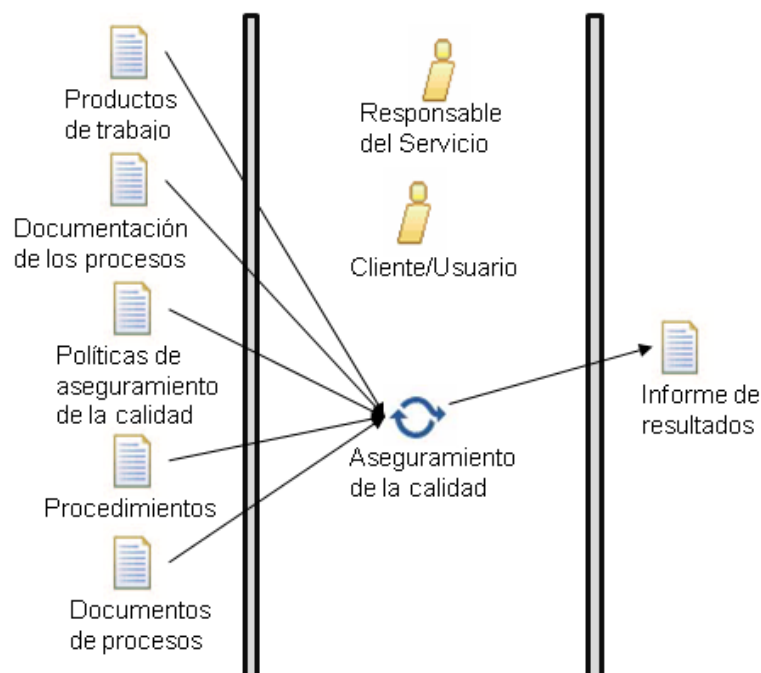
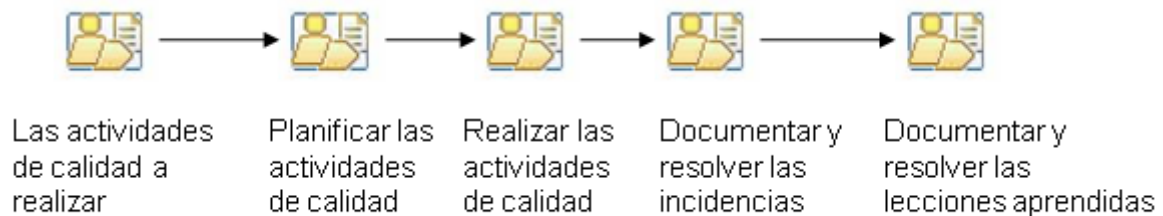


Figura 53: Proceso Aseguramiento de la Calidad

Es necesario verificar que cada una de las actividades que han sido definidas en los procesos implantados en la organización se está llevando a cabo y que los productos intermedios y finales que se obtienen son adecuados.

En la figura 54, se muestran las distintas actividades que definen el proceso.



**Figura 54: Actividades del proceso Aseguramiento de la Calidad**

### **Actividad 1: Las actividades de calidad a realizar**

El objetivo principal de esta actividad es identificar y determinar el conjunto de actividades de Aseguramiento de la Calidad a realizar, tomando como base la Documentación de los procesos, las Políticas de aseguramiento de calidad vigentes en la organización y el Catálogo de servicios disponibles.

Para poder determinar las actividades de calidad será necesario previamente seleccionar el conjunto de procesos que se van a considerar como aplicables. Una vez se haya realizado esta selección, se podrán determinar los mecanismos que van a ser utilizados para verificar su correcta aplicación en el ámbito del servicio.

Cada uno de los procesos tendrá asociado un conjunto de ítems que permiten verificar su correcta aplicación en un servicio. Estos ítems se denominan elementos de verificación. Así, a partir de los procesos seleccionados para el proyecto de las Listas de Verificación y de la información incluida en el catálogo del servicio, se determinarán los elementos de verificación que se deben controlar para asegurar la calidad del servicio en curso. Asimismo, también se deberá indicar qué elementos de verificación se excluyen de las actividades de calidad configurando las listas de verificación de cada proceso aplicable en un proyecto determinado.

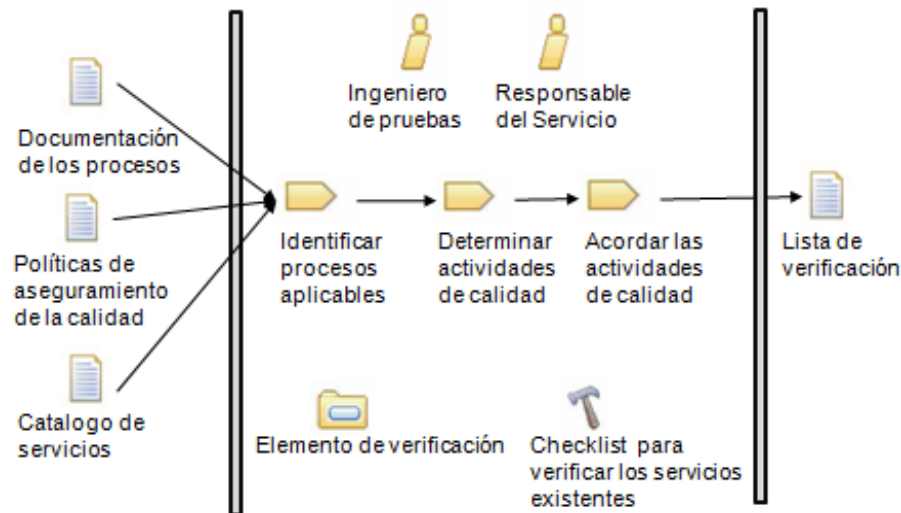


Figura 55: Actividad 1: Las actividades de calidad a realizar

## Actividad 2: Planificar las actividades de calidad

El objetivo principal de esta actividad es realizar la planificación de las actividades de calidad que han sido previamente acordadas tomando como base la lista de verificación que contiene las actividades aplicables que han sido acordadas.

Para realizar las actividades de aseguramiento de calidad de una forma ordenada y controlada es necesario planificarlas, teniendo en cuenta la ejecución de los servicios afectados. Se realiza la asignación de recursos y de responsables y se elabora el calendario de realización de las mismas.

Básicamente la realización de cada una de estas actividades consiste en cumplimentar adecuadamente las listas de verificación acordadas en función de los hallazgos encontrados.

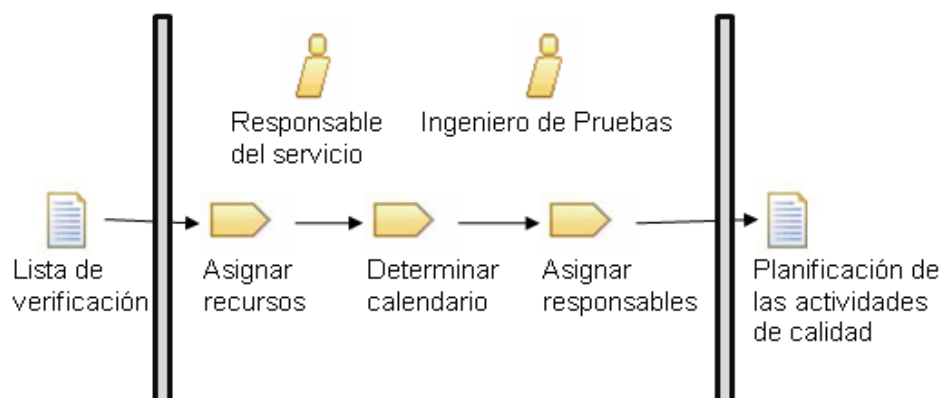


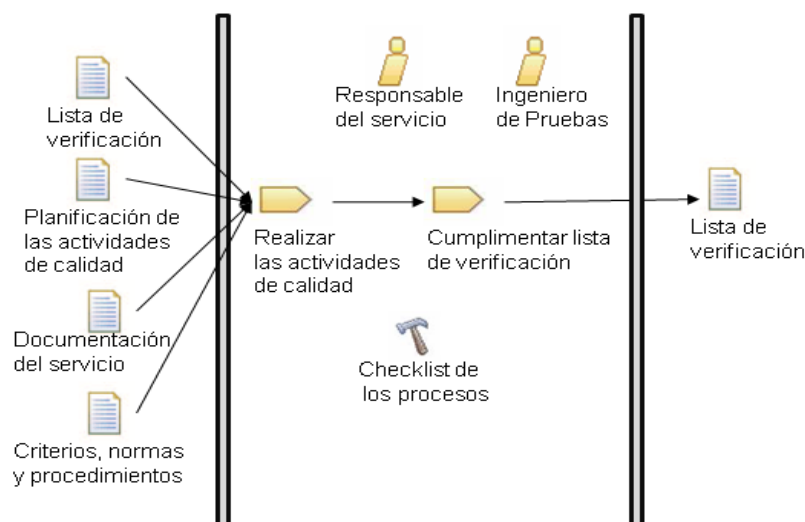
Figura 56: Actividad 2: Planificar las actividades de calidad

### **Actividad 3: Realizar las actividades de calidad**

El objetivo principal de esta actividad es llevar a cabo cada una de las actividades de calidad que han sido planificadas previamente, tomando como base la lista de verificación que contiene las actividades aplicables que han sido acordadas así como la planificación de las actividades de calidad, la documentación relacionada con el servicio y los criterios, normas y procedimientos que son necesarios para realizar las actividades de calidad.

Los responsables que hayan sido asignados para realizar las actividades de calidad, las llevarán a cabo en las fechas previstas y de acuerdo con las normas y técnicas de trabajo acordadas.

Para ello, cada uno de responsable deberá cumplimentar las listas de verificación a medida que dichas actividades se vayan realizando o justificando debidamente aquellas actividades comprometidas y que no se hayan realizado.



**Figura 57: Actividad 3: Realizar las actividades de calidad**



#### Actividad 4: Documentar y resolver las incidencias

El objetivo principal de esta actividad es documentar y resolver las incidencias que surjan durante el desarrollo del servicio como consecuencia de que o bien no se ha realizado una actividad de un proceso definido o bien porque ha sido realizada con métodos distintos a los especificados en dichos procesos, tomando como base la lista de verificación documentada.

Las incidencias identificadas durante el seguimiento de las actividades de calidad serán cumplimentadas describiendo cómo, quién y cuándo deben ser resueltas. Además, las acciones recomendadas deberán ser seguidas y controladas, realizándose el cierre de las mismas cuando hayan sido realizadas correctamente, indicando la fecha de cierre y dejando constancia de ello.



Figura 58: Actividad 4: Documentar y resolver las incidencias

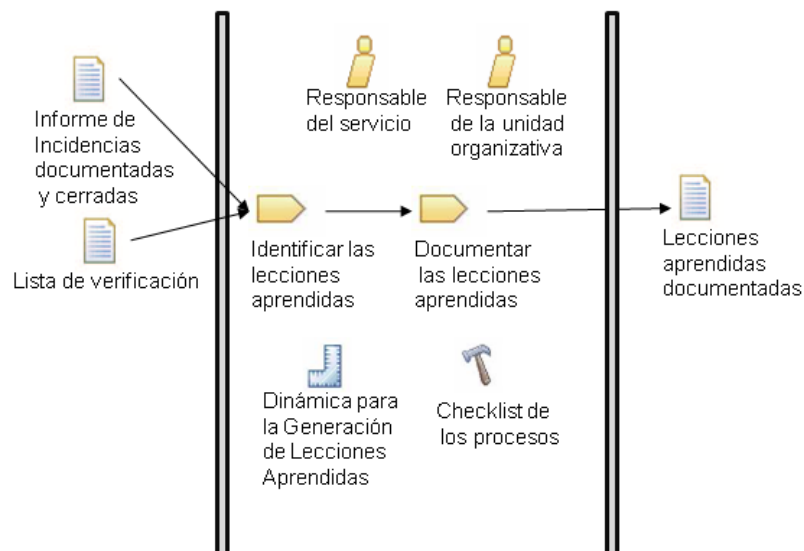
#### Actividad 5: Documentar y resolver las lecciones aprendidas

El objetivo principal de esta actividad es documentar y resolver las lecciones aprendidas que se hayan obtenido durante la ejecución del servicio.

Las lecciones aprendidas serán las prácticas excelentes y las prácticas deficientes que han sido realizadas durante el desarrollo del proyecto. Éstas podrán ser extraídas a partir de incidencias que han sido identificadas y documentadas, mediante la realización de dinámicas de grupo.

Para ello, al finalizar el proyecto, se realizará una reunión con el equipo de proyecto en la que, para cada área de proceso, se deberán identificar las prácticas excelentes, prácticas deficientes y riesgos altos para posteriormente documentar las lecciones aprendidas, detallando el área de proceso al que referencian y las personas involucradas.

En esta reunión, además del equipo de trabajo y el responsable del servicio, puede participar el Responsable de la unidad organizativa.



**Figura 59: Actividad 5: Documentar y resolver las lecciones aprendidas**

### **3.4 Conclusiones**

En este capítulo, han sido definidos cada uno de los procesos de Verificación y Validación que hay que realizar para asegurar la calidad en los productos. Por cada uno de estos procesos, se observa cual es el orden de las actividades a realizar y dentro de cada actividad se especifica el orden de las tareas a realizar además de conocer cuáles son las entradas y salidas dentro de cada una de las actividades. También se definen los roles responsables de cada proceso y los elementos de trabajo, instrucciones técnicas y productos de trabajo necesarios para desempeñar cada actividad. Una vez los procesos han sido definidos, en el siguiente capítulo se procede a describir como se ha llevado a cabo la implementación de dicho modelo mediante el uso de la EPG (Guía Electrónica de Procesos) y que por tanto constituye nuestro Modelo de Referencia.

## CAPÍTULO 4

# IMPLEMENTACIÓN DEL MODELO

#### **4. IMPLEMENTACIÓN DEL MODELO**

Tanto los nuevos procesos que se definan cómo las mejoras que se planteen deben ser guiados de manera efectiva para conseguir que los progresos se hagan palpables [49].

A partir de esta premisa, convertida en objetivo fundamental de los manuales impresos sobre procesos podemos comenzar esta primera aproximación a las guías de proceso.

Pero, ¿qué entendemos por Proceso Software?: Un Proceso Software (PS) es “un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software” [50].

¿Y una guía de proceso?: “Una guía de proceso es un documento de referencia para un proceso particular que proporciona asistencia a los equipos de trabajo que participan en el desarrollo de dicho proceso” [51]. Contienen básicamente definiciones de procesos y proporcionan orientación.

Una guía de procesos [52] [53] pretende describir un proceso particular con el propósito de dar soporte a las personas encargadas de la realización de dicho proceso, por tanto, los equipos de trabajo que participan en el proceso son los usuarios a los que van dirigidas las guías de procesos. Es un documento de referencia estructurado y orientado a flujos de trabajo, asociado a un proceso particular.

En base a lo anterior, podemos establecer una primera aproximación sobre los beneficios más relevantes obtenidos al usar una guía de procesos [51]:

- Facilitan la comunicación tanto online como offline: la comunicación online podría darse entre las personas que forman parte de los equipos de trabajo que participan en el proceso que llevan a cabo acciones relacionadas mientras que la comunicación offline podría darse entre autores y lectores de documentos de procesos.
- Ayudan a los equipos de trabajo que participan en el proceso a trazar su trabajo en base a información sobre eventos del proceso (por ejemplo, tan sencillo como tachar tareas una vez que han sido completadas).
- Ayudan a los equipos de trabajo que participan en el proceso a realizar su trabajo eficientemente (por ejemplo, en base a guías que hagan referencia a experiencias que ayuden a elegir entre diferentes alternativas en una toma de decisiones).
- Ayudan a los equipos de trabajo que participan en el proceso a volver a encarar un proceso, completo o en parte (por ejemplo, información del estado podría ayudarles a entender lo que necesitan para reiniciar un trabajo después de una interrupción).

Sin embargo, estas ventajas no siempre son aplicables, a pesar de que las guías impresas, los estándares, los manuales sobre procesos y procedimientos están ampliamente utilizados en la industria, ya que frecuentemente estos documentos no son vistos como útiles por sus eventuales lectores.

A continuación se incluyen sus principales problemas que afectan a estas guías:

- Los lectores no pueden navegar fácilmente a través de las páginas si no siguen el flujo de información del documento.
- Falta de información clave en las guías tradicionales.
- Las anotaciones y aclaraciones incluidas en las guías por un lector no pueden ser compartidas con otros.
- El control de versiones no está soportado de la mejor manera y además la distribución de una nueva versión de un proceso a sus participantes es poco formal.
- Las guías o bien contienen información mezclada para diferentes lectores, o bien están compuestas de documentos orientados a necesidades específicas que sin embargo comparten una descripción de proceso común.
- Las guías no están diseñadas para almacenar información de estado del proyecto.

Nos encontramos pues con dos ideas aparentemente enfrentadas: los beneficios de la utilización de las guías de procesos frente al fracaso de las guías de procesos escritas.

“La función central de cualquier guía de procesos es facilitar el entendimiento del proceso” [51]. A partir de esta afirmación, y en vista de la contraposición que se acaba de plantear, surge la idea de que esta función principal puede ser ejecutada de manera más efectiva empleando las tecnologías de la información en lugar de las guías escritas en papel.

Como parte de la evolución hacia la digitalización en la sociedad en la que vivimos, los manuales y guías que se utilizan para orientar a sus respectivos lectores está siendo generados y consultados cada vez más en formato electrónico, de ahí el origen del concepto de “Guía Electrónica de Procesos”, que responde en inglés a las siglas de EPG (Electronic Process Guide) y que será el acrónimo que utilizaremos de aquí en adelante para referirnos a dicho concepto. [54]

Una Guía Electrónica de Procesos facilita la ejecución de los procesos ayudando a la realización de sus actividades, indicando cómo y cuándo realizarlas. Para cada uno de los procesos se incorpora una serie de elementos como son: Instrucciones técnicas para la realización de las actividades de los procesos a nivel técnico y metodológico, plantillas para la generación de documentos de gestión de proyectos y una descripción detallada de los elementos de trabajo involucrados en el desarrollo de actividades.[55] Toda EPG deberá cumplir los siguientes requerimientos que se enumeran a continuación:

- Proporcionar las unidades básicas de información, es decir, actividades, artefactos, roles, agentes y recursos, así como las principales relaciones entre ellos.
- Proporcionar todos los elementos de información y sus respectivas interrelaciones contenidos en una buena guía de procesos en papel, es decir, la información no debe perderse en el procedimiento de traslado desde su ubicación física en papel hacia su ubicación en formato digital.
- Utilizar ampliamente y de modo efectivo los hipervínculos para dar soporte a una navegación flexible a través de la información contenida en la guía. Las referencias a documentos internos o externos deberían estar asociadas con hipervínculos, de

modo que siguiendo dichos hiperenlaces, el usuario pudiera acceder directamente a la información correspondiente.

- Acceder fácilmente a la información deseada. Los usuarios deberían poder acceder a información frecuentemente utilizada de manera rápida. Debería ser posible conseguir información sin necesidad de navegar a través de otras actividades.
- Las páginas web deberían tener la misma estructura básica o al menos similar a la EPG para facilitar la orientación a los usuarios.
- El usuario no debe ser sobrecargado con demasiadas ventanas, el número de ventanas debe ser limitado y bien gestionado. Los usuarios deben tener el control sobre la apertura, cierre, redimensionamiento y reposicionamiento de las ventanas.
- La tecnología a emplear, referida a gestores de ventanas y navegadores, debe ser conocida por los usuarios, para no acarrearles ningún problema tecnológico importante.

La estructura de una EPG debe de ser idéntica a la estructura que poseen los procesos definidos. La estructura que presentan los procesos y por tanto la EPG ha sido definida en la sección 3.2 (Especificación de los procesos) del presente documento.

La documentación final del proceso se adaptó al formato de Guía Electrónica. Para ello, se paso la documentación a un formato Web que diese lugar a la guía electrónica. Se partió de una serie de plantillas desarrolladas con Microsoft InfoPath, aplicación que permite desarrollar formularios de entrada de datos basados en XML que posteriormente fueron rellenados de manera sencilla y que sirvieron de base para cada uno de los procesos, actividades, elementos de trabajo e instrucciones técnicas elaboradas. Una vez rellenas las plantillas se generan los ficheros XML correspondientes a cada uno de los procesos definidos así como cada una de las actividades que componen dicho proceso. Cada uno de estos ficheros XML son subidos al servidor de SourceSafe, siendo SourceSafe una herramienta que permite el control de versiones basadas en Cliente-Servidor donde el repositorio de control de cambios reside en el equipo servidor y los clientes toman de allí la última versión para modificarla y posteriormente subirla con las modificaciones realizadas. Como en todos los programas de control de versiones, se basa en obtener una copia de trabajo ("check-out" o "desproteger"), realizar cambios sobre la copia y reingresarla al repositorio. Por ello, toda la información del proyecto está almacenada en dicho servidor.

A continuación se muestra un ejemplo de la implementación en la EPG de uno de los procesos que forman parte del Modelo de Procesos de Verificación y Validación. Se trata del proceso de **Verificación de los Productos de Trabajo**:

Portal SEL >

**Procesos**

- Actividades
- Elementos de Trabajo
- Productos de Trabajo
- Instrucciones Técnicas

**Verificación de Productos de Trabajo**

**Ingeniero de Pruebas**

**Visión General**

El objetivo principal del proceso de verificación de productos de trabajo es asegurar que aquellos productos que intervienen en el proceso de desarrollo de software son correctos, consistentes y completos, es decir que satisfacen las necesidades del cliente. Para ello, se llevará a cabo una revisión del contenido del producto de trabajo así como de los procedimientos o actividades establecidos para su gestión.

El principal motivo de llevar a cabo una verificación de productos de trabajo es evitar que éstos se traduzcan en defectos en el producto final. Es decir, un defecto en los requisitos se traducirá en un defecto en el sistema que se desarrolle.

Los principales beneficios obtenidos de revisar productos de trabajo son:

- Incremento de la calidad del producto como consecuencia de la eliminación de defectos.
- Reducción del re-trabajo y del esfuerzo, la detección temprana de errores reduce el re-trabajo y esfuerzo en fases de desarrollo posteriores.
- Disminución del número de re-pruebas como consecuencia de la disminución de los fallos encontrados durante las pruebas.

Por tanto, las revisiones son un método efectivo para la detección de errores y discrepancias que conlleva un incremento de la fiabilidad del producto, un impacto positivo en el calendario y una reducción de los costes.

Existen diferentes técnicas para llevar a cabo una revisión, estableciendo una distinción entre revisiones formales o informales.

**Roles Participante**

Responsables:  
**Ingeniero de Prueba**

Informados:  
**Responsable del Servicio Cliente/Usuario**

Figura 60:EPG.Visión General: Proceso Verificación de Productos de Trabajo

Como podemos observar en la figura 59, lo primero que aparece es la visión general del proceso en la que se explican los diferentes objetivos y beneficios de llevar a cabo dicho proceso así como los Roles Participantes. A continuación en la figura 60, podemos ver cuáles son los Criterios de Entrada necesarios y en qué momento se debe llevar a cabo el proceso. Después conocemos que actividades componen dicho proceso, cuál es el objetivo de cada una y en qué orden se deben realizar.

### Criterios de Entrada

- **Paquete de Documentación del Proyecto**  
Incluye la documentación que se ha elaborado sobre el proyecto al que pertenecen el producto de trabajo a verificar (especificaciones de requisitos, documento de análisis, documento de diseño, plan de proyecto, o cualquier otra especificación que se haya realizado).
- **Producto de Trabajo**  
Producto de trabajo que se va a verificar.
- **Proceso de construcción del producto de trabajo**  
Especificación del proceso que se ha utilizado para el desarrollo de trabajo que se va a verificar.
- **SLA**  
Acuerdo de nivel de servicio.

#### Cuándo:

- Una vez se haya firmado un acuerdo de servicio con el cliente.

### Actividades

- |   |                                       |  |
|---|---------------------------------------|--|
| 1 | Seleccionar los proyectos a verificar | El objetivo de esta actividad es seleccionar el conjunto de proyectos sobre los que se desea aplicar el proceso de verificación.   |
| 2 | Planificar el proceso de revisión     | Esta actividad tiene como objetivo realizar la planificación del proceso de revisión.  |
| 3 | Realizar Inspección                   | El objetivo principal de esta actividad es realizar la inspección de productos o entregables de un proyecto.   |
| 4 | Realizar Walkthroughs                 | El objetivo principal de esta actividad es llevar a cabo una revisión informal o walkthroughs con el objetivo de obtener nuevas soluciones a un problema determinado   |
| 5 | Verificar procedimientos de gestión   | Esta actividad tiene como objetivo revisar las actividades y los procedimientos que han sido realizados para la obtención del producto de trabajo revisado, con el propósito de asegurar que el proceso implantado se está realizando correctamente. |
| 6 | Elaborar informe de resultados        | Esta actividad tiene como objetivo elaborar el informe final que recoge los resultados de la realización del proceso de verificación.  |

Figura 61:EPG.Actividades del Proceso Verificación de Productos de Trabajo

Por último, una vez llevado a cabo las actividades del proceso, se obtiene la correspondiente salida siendo para este proceso de Verificación y Validación la siguiente:

### Criterios de Salida

- ✓ Informe de resultados

Figura 62:EPG.Criterios de Salida del proceso Verificación de Productos de Trabajo

Cabe destacar que si seleccionamos cualquiera de las actividades que componen el proceso de Verificación y Validación de productos de Trabajo, podemos ver desde dentro la estructura de dicha actividad. En la figura 62 se muestra la estructura de la actividad 1 de dicho proceso:





Figura 63:EPG.Actividad1 del proceso Verificación de los Productos de Trabajo

Dentro de esta actividad, se puede ver la visión general de dicha actividad y por tanto conocer cuáles son sus objetivos así como las entradas necesarias para llevarla a cabo y los Roles Participantes.

Por último, en la figura 63 se muestran el orden y en qué consiste cada una de las tareas a realizar dentro de la actividad así como cual es la salida obtenida una vez han sido realizadas las tareas y por tanto se puede dar por finalizada la actividad.

#### Tareas

- |          |                           |  |
|----------|---------------------------|--|
| <b>1</b> | Identificar los proyectos | Identificar los proyectos que se van a utilizar en el proceso de verificación, en función de un conjunto de criterios previamente definidos. |
| <b>2</b> | Identificar el orden      | Establecer el orden según el cual se van a verificar los productos de trabajo de los proyectos seleccionados previamente.                    |

#### Criterios de Salida

- ✓ Lista de proyectos seleccionados

Figura 64:EPG.Tareas de la Actividad 1 del proceso Verificación de los Productos de Trabajo

## CAPITULO 5

## CONCLUSIONES Y LÍNEAS FUTURAS

## **5. CONCLUSIONES Y LÍNEAS FUTURAS**

### **5.1 Conclusiones**

En la actualidad la satisfacción respecto a la calidad obtenida de un producto es primordial debido a que marca la diferencia en el mercado en cuanto a productos similares. Insistir en el buen desarrollo de los productos provoca que las expectativas de los clientes se vean satisfechas y marcará la diferencia entre dos organizaciones que compiten en el mismo.

Hoy en día el concepto de calidad es un término imprescindible en cualquier ámbito y en el cual nos aferramos a la hora de tomar decisiones respecto a un producto ya que es un concepto que nos transmite seguridad a la hora de conocer si ese producto satisface todos nuestros intereses.

Todos conocemos que son numerosos los problemas de calidad que presentan los productos software debido a un mal desarrollo del producto. Seguir los procesos adecuados para realizar un buen desarrollo de un producto como son los planteados en este proyecto, conlleva a obtener productos de alta calidad que cumplirán todas las expectativas planteadas.

Por todo ello, sería interesante para las organizaciones contar con un Modelo de Referencia como el planteado en este proyecto, ya que está basado en procesos compuestos por diversas actividades que se ocupan de verificar y validar que todos los productos finales ofrecidos al cliente son de calidad y garantizar que los problemas de calidad se vean reducidos o desaparezcan.

El estudio de modelos existentes ha sido costoso puesto que es un tema con un gran volumen de información y ha sido laborioso componer un modelo de referencia propio que garantice la eficiencia y eficacia de los productos.

Finalmente la elaboración de todos los procesos que conforman el modelo aquí expuesto ha sido una labor compleja, pero gratificante, puesto que una vez que se han fijado las bases de este modelo se puede ver cómo puede ser aplicado no sólo a la validación y verificación de productos software, sino también a otros productos que se quiera garantizar su calidad.

A título personal, me siento totalmente satisfecha por el trabajo realizado en este proyecto y el cuál ha sido posible gracias a la inestimable ayuda de los tutores del proyecto. Así he conseguido elaborar un Modelo de Referencia que puede servir de ayuda a cualquier organización que desee mejorar la calidad en sus productos software.

## **5.2 Líneas Futuras**

Finalmente, se recogen un conjunto de ideas surgidas durante el análisis y desarrollo del proyecto que podrían tenerse en cuenta a la hora de evolucionar la solución presentada:

- Se podrían documentar otros procesos para asegurar la calidad en los productos en base a los propuestos en el modelo de procesos del apartado 3.
- Se podría adaptar el modelo de procesos a organizaciones más grandes que implementen software más complejo.
- Se podrían añadir cuestionarios para recabar información de los equipos de trabajo que participan en el proceso para tratar de mejorar la guía en base a sus opiniones y recomendaciones.
- Se podría documentar que otros procesos a parte de las pruebas software se podrían externalizar en una organización con el fin de modelar su trabajo pero manteniendo la calidad en sus productos.
- En cuanto al modelo de referencia se podría introducir más instrucciones técnicas que detallen los procedimientos a seguir en determinadas actividades, resultando de mayor ayuda al usuario.

## CAPITULO 6

# PLANIFICACIÓN Y PRESUPUESTO

## 6. PLANIFICACIÓN Y PRESUPUESTO

### 6.1 Planificación

En este apartado se muestra cuál hubiera sido la planificación inicial del proyecto sino se hubiera sufrido ningún tipo de retraso así como la planificación que realmente se ha llevado a cabo.

#### Planificación inicial del proyecto:

Nombre de tarea	Duración	Comienzo	Fin
Inicio del proyecto	374 días	mar 01/09/09	mar 01/02/11
[-] Identificar documentación relevante	66 días	mar 01/09/09	lun 30/11/09
Búsquedas en internet sobre calidad, aseguramiento de la calidad y gestión de la configuración en los productos software	22 días	mar 01/09/09	mié 30/09/09
Lectura de libros de ingeniería de software sobre pruebas software, outsourcing, verificación y validación	22 días	jue 01/10/09	sáb 31/10/09
Recibir información de los tutores del proyecto acerca de que es un modelo de referencia y un modelo de procesos	22 días	dom 01/11/09	lun 30/11/09
[-] Analizar documentación	64 días	mar 01/12/09	dom 28/02/10
Lectura detallada de la información para saber si era información útil o no en mi proyecto	44 días	mar 01/12/09	dom 31/01/10
Asesoramiento por parte de los tutores del proyecto acerca de que información iba a ser tratada en el proyecto	1 día	vie 26/02/10	dom 28/02/10
[-] Diseñar los procesos	67 días	lun 01/03/10	lun 31/05/10
Documento que recoge los conceptos encargados de determinar que un producto es de calidad	23 días	lun 01/03/10	mié 31/03/10
Traducir los conceptos que aseguran la calidad de un producto a procesos	1 día	vie 30/04/10	vie 30/04/10
Esquema que contiene los procesos encargados de asegurar que un producto es de alta calidad	22 días	sáb 01/05/10	lun 31/05/10
[-] Definir los procesos	67 días	mar 01/06/10	mar 31/08/10
Identificar las actividades que componen el proceso	22 días	mar 01/06/10	mié 30/06/10
Identificar las tareas que componen cada una de las actividades del proceso	1 día	vie 30/07/10	sáb 31/07/10
Identificar las entradas y salidas del proceso	23 días	dom 01/08/10	mar 31/08/10
Documento por cada proceso que recoja toda la información anterior	22 días	lun 02/08/10	mar 31/08/10
[-] Seguimiento de los procesos	44 días	mié 15/09/10	lun 15/11/10
[-] Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos	44 días	mié 15/09/10	lun 15/11/10
Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos 1	1 día	mié 15/09/10	mié 15/09/10
Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos 2	1 día	vie 15/10/10	vie 15/10/10
Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos 3	1 día	lun 15/11/10	lun 15/11/10
[-] Evaluar el proyecto	483 días?	mié 01/12/10	vie 05/10/12
Lectura del proyecto por parte de los tutores	23 días	mié 01/12/10	vie 31/12/10
Lectura del proyecto por parte del autor del mismo	1 día	lun 31/01/11	lun 31/01/11

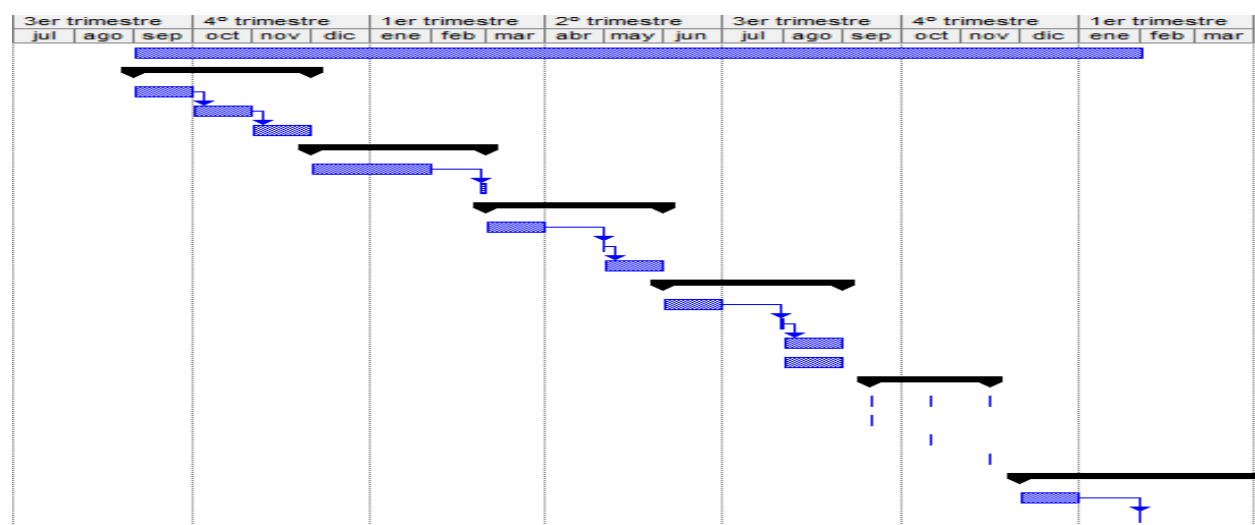


Figura 65: Planificación inicial del proyecto

## Planificación real del proyecto:

Nombre de tarea	Duración	Comienzo	Fin
Inicio del proyecto	461 días?	mar 01/09/09	mar 31/05/11
[-] Identificar documentación relevante	89 días?	mar 01/09/09	jue 31/12/09
Búsquedas en internet sobre calidad, aseguramiento de la calidad y gestión de la configuración en los productos software	44 días?	mar 01/09/09	sáb 31/10/09
Lectura de libros de ingeniería de software sobre pruebas software, outsourcing, verificación y validación	22 días?	dom 01/11/09	lun 30/11/09
Recibir información de los tutores del proyecto acerca de que es un modelo de referencia y un modelo de procesos	23 días?	mar 01/12/09	jue 31/12/09
[-] Analizar documentación	86 días?	vie 01/01/10	vie 30/04/10
Lectura detallada de la información para saber si era información útil o no en mi proyecto	41 días?	vie 01/01/10	dom 28/02/10
Asesoramiento por parte de los tutores del proyecto acerca de que información iba a ser tratada en el proyecto	1 día?	vie 30/04/10	vie 30/04/10
[-] Diseñar los procesos	89 días?	sáb 01/05/10	mar 31/08/10
Documento que recoge los conceptos encargados de determinar que un producto es de calidad	44 días?	sáb 01/05/10	mié 30/06/10
Traducir los conceptos que aseguran la calidad de un producto a procesos	1 día?	vie 30/07/10	sáb 31/07/10
Esquema que contiene los procesos encargados de asegurar que un producto es de alta calidad	23 días?	dom 01/08/10	mar 31/08/10
[-] Definir los procesos	88 días?	mié 01/09/10	vie 31/12/10
Identificar las actividades que componen el proceso	22 días?	mié 01/09/10	jue 30/09/10
Identificar las tareas que componen cada una de las actividades del proceso	1 día?	vie 29/10/10	dom 31/10/10
Identificar las entradas y salidas del proceso	22 días?	lun 01/11/10	mar 30/11/10
Documento por cada proceso que recoja toda la información anterior	22 días?	jue 02/12/10	vie 31/12/10
[-] Seguimiento de los procesos	42 días	lun 17/01/11	mar 15/03/11
+ Lectura iterativa de los procesos con el fin de añadir, eliminar o modificar cualquier información relativa a ellos	42 días	lun 17/01/11	mar 15/03/11
[-] Evaluar el proyecto	44 días?	vie 01/04/11	mar 31/05/11
Lectura del proyecto por parte de los tutores	21 días?	vie 01/04/11	sáb 30/04/11
Lectura del proyecto por parte del autor del mismo	23 días?	dom 01/05/11	mar 31/05/11

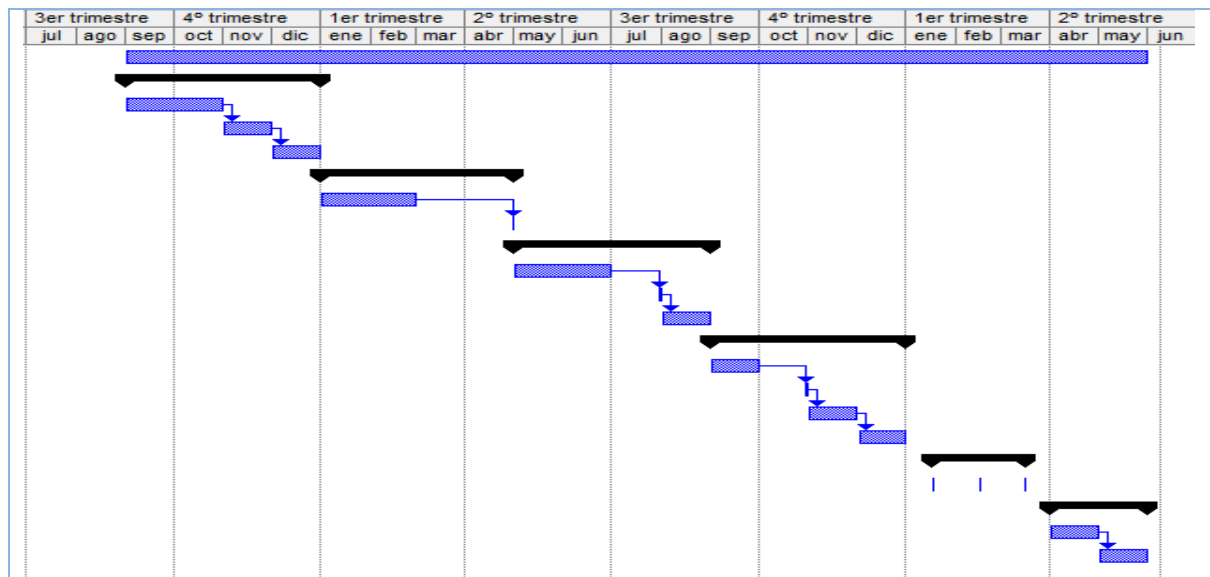


Figura 66: Planificación real del proyecto

## 6.2 Presupuesto

A continuación se ofrece un presupuesto básico del coste aproximado estimado para poder realizar este proyecto.


 <b>UNIVERSIDAD CARLOS III DE MADRID</b> <b>Escuela Politécnica Superior</b>							
<b>PRESUPUESTO DE PROYECTO</b>							
<b>1 - Autor:</b>							
Jessica Castellanos Fraga							
<b>2 - Departamento:</b>							
Laboratorio de Ingeniería del Software							
<b>3 - Descripción del Proyecto:</b>							
- Título	Especificación e implementación de un modelo de provisión de servicios para Validación y Verificación de productos sw						
- Duración (meses)	21						
Tasa de costes indirectos:	20%						
<b>4 - Presupuesto total del Proyecto (valores en Euros):</b>							
Euros							
<b>5 - Desglose presupuestario (costes directos)</b>							
<b>PERSONAL</b>							
Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (meses) <sup>a)</sup>	(hombres)	Coste hombre mes	Coste (Euro)	Firma de conformidad
Jessica Castellanos Fraga	0000000-A	Ingeniero Junior		1575	1.500,00	31.500,00	-----
Sanz Esteban, Ana	0000000-B	Ingeniero Senior		1155	2.800,00	58.800,00	-----
Saldaña Ramos, Javier	0000000-C	Ingeniero Senior		1155	2.800,00	58.800,00	-----
						0,00	
<b>Hombres mes 3885</b>					<b>Total</b>	<b>149.100,00</b>	
<sup>a)</sup> 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas) Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)							
<b>EQUIPOS</b>							
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Período de depreciación	Coste imputable <sup>a)</sup>		
PC Portatil Asus A52JU-SX036 Po	599,00	100	12	60	119,80		
PC Portatil Asus A52JU-SX036 Po	599,00	100	12	60	119,80		
PC Portatil Asus A52JU-SX036 Po	599,00	100	12	60	119,80		
Servidor Alojjar Web DELL™ Powe	1.300,00	100	12	60	260,00		
		100		60	0,00		
					0,00		
					<b>Total</b>	<b>619,40</b>	
<sup>a)</sup> Fórmula de cálculo de la Amortización: $\frac{A}{B} \times C \times D$ <p>A = nº de meses desde la fecha de facturación en que el equipo es utilizado  B = periodo de depreciación (60 meses)  C = coste del equipo (sin IVA)  D = % del uso que se dedica al proyecto (habitualmente 100%)</p>							
<b>SUBCONTRATACIÓN DE TAREAS</b>							
Descripción	Empresa	Coste imputable					
No aplicable							
		<b>Total</b>	<b>0,00</b>				
<b>OTROS COSTES DIRECTOS DEL PROYECTO<sup>a)</sup></b>							
Descripción	Empresa	Costes imputable					
Papelería y material oficina	Office Paper	600,00					
		<b>Total</b>	<b>600,00</b>				
<sup>a)</sup> Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas,							
<b>6 - Resumen de costes</b>							
Presupuesto Costes Totales	<b>Presupuesto Costes Totales</b>						
Personal	149.100						
Amortización	619						
Subcontratación de tareas	0						
Costes de funcionamiento	600						
Costes indirectos	30.064						
<b>Total</b>	<b>180.383</b>						

Figura 67: Presupuesto



## CAPÍTULO 7

## GLOSARIO

## **7. GLOSARIO**

- **Aseguramiento de Calidad (QA):** Son todas las actividades planificadas y sistemáticas que son necesarias para proporcionar la confianza adecuada de que un producto o servicio cumplirá los requisitos dados sobre la calidad.
- **Aseguramiento de la Calidad Software (SQA):** Son todas las actividades planificadas que son necesarias para proporcionar la confianza de que un producto software cumplirá los requisitos de calidad.
- **CMMi:** Se define CMMI como modelo de referencia que engloba el desarrollo y el mantenimiento tanto de productos como de servicios software. En él, se desarrollan prácticas que abarcan la gestión del proyecto, la gestión de procesos, la ingeniería de sistemas, la ingeniería del software, la ingeniería del hardware y otros procesos de soporte utilizados en el desarrollo y el mantenimiento.
- **Elemento de configuración (EC):** Elemento de información que se crea como parte del proceso de desarrollo de un producto o servicio.
- **EPG (Electronic Process Guide):** “Guía Electrónica de Procesos”. Es una guía que facilita la ejecución de los procesos ayudando a la realización de sus actividades, indicando cómo y cuándo realizarlas.
- **Gestión de Configuración (GC):** Disciplina que identifica la configuración de un sistema en determinados puntos del tiempo. Es un conjunto de actividades de seguimiento y control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento que comienzan cuando se inicia el proyecto de desarrollo de un producto y termina sólo una vez que el producto queda fuera de circulación.
- **ISO/IEC 12207 (International Organization for Standardization):** Information Technology / Software Life Cycle Processes, es el estándar para los procesos de ciclo de vida del software de la organización ISO. Establece un proceso de ciclo de vida que incluye procesos y actividades que se aplican desde la definición de requisitos, pasando por la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso.
- **ISO 9000 (International Organization for Standardization):** Designa un conjunto de normas sobre calidad y gestión continua de calidad, establecidas por ISO. Se pueden aplicar en cualquier tipo de Organización o actividad orientada a la producción de bienes o servicios. Las normas recogen tanto el contenido mínimo como las guías y

herramientas específicas de implantación, como los métodos de auditoría. El ISO 9000 especifica la manera en que una organización opera, sus estándares de calidad, tiempos de entrega y niveles de servicio.

- **NPR (número de prioridad de riesgo):** Es un valor que indica el nivel de prioridad que tiene un riesgo o la frecuencia de ocurrencia, la gravedad y el grado de facilidad para su detección. Mediante ese valor se priorizan los fallos identificados.
- **Plan de Gestión de Configuración (PGC):** Es un plan en el que se plasmará todas aquellas actividades que se deben llevar a cabo para mantener una buena gestión y organización del desarrollo y mantenimiento del producto.
- **Plan de Aseguramiento de Calidad (PQA):** Es un plan mediante el cual se plasmarán todas las actividades necesarias para llevar a cabo un buen desarrollo de un producto o servicio software haciendo que este goce de una buena calidad.
- **RTM (Release To Market):** Es un proceso mediante el cual se comprueba la funcionalidad del programa completo en entornos de producción.
- **SPICE (Software Process Improvement and Capability Determination):** Sistema de calidad de productos software que permite la evaluación de procesos software en las organizaciones.
- **TMM: (Testing Maturity Model, TMM):** Su propósito es servir como una guía para realizar una mejora del proceso de pruebas además de servir como complemento a CMM. TMM al igual que CMM utiliza el concepto de niveles de madurez para la evaluación y la mejora de los procesos.
- **TMMi (Testing maturity model integrated (TMMi)):** Es la evolución de TMM, desarrollada por “TMMi foundation” para la mejorar del proceso de pruebas. Dicho modelo surgió como respuesta y complemento de CMMi, ya que su enfoque, al ser más amplio y estructurado, logra un impacto favorable sobre la calidad del producto.

## CAPÍTULO 8

## REFERENCIAS

## **8. REFERENCIAS**

- [1] ISO 9000 (International Organization for Standardization), (1987)
- [2] Jose Luis Penachob (Ingeniero Industrial), Evolución Histórica de la Calidad en el contexto del mundo de la empresa y del trabajo, (2000)  
[http://www.forumcalidad.com/pdf/Series\\_FC116\\_penacho.pdf](http://www.forumcalidad.com/pdf/Series_FC116_penacho.pdf)
- [3] Christian Alejandro Isidoro Nambo, Instituto Tecnológico Superior de Apatzingan, Conceptos Filosofía de Calidad.  
<http://www.mitecnologico.com/Main/ConceptosFilosof%EDasDeCalidad>
- [4] César González Arévalo, Calidad y Gestión de la Calidad (2007)  
<http://www.gestiopolis.com/canales8/ger/calidad-por-edwards-deming.htm>
- [5] Gestión de la Calidad y BPA, La calidad total de Ishikawa  
<http://bpa.peru-v.com/ishikawa.htm>
- [6] Mc Graw Hill "Quality is Free", New York, (1979)
- [7] Evolución Calidad  
<http://www.mitecnologico.com/Main/EvolucionCalidad>
- [8] Wikipedia, la enciclopedia libre
- [9] Metodología MÉTRICA Versión 3, Ministerio de Política Territorial y Administración Pública  
[http://administracionelectronica.gob.es/?\\_nfpb=true&\\_pageLabel=P60085901274201580632&langPae=es](http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es) [10] [www.eumed.net](http://www.eumed.net)
- [10] Beneficios de los procesos de Aseguramiento de la Calidad en el Software  
<http://www.eumed.net/libros/2008a/351/Beneficios%20de%20los%20procesos%20de%20Aseguramiento%20de%20la%20Calidad%20en%20el%20Software.htm>
- [11] Propuesta de actividades del proceso de Aseguramiento de la Calidad del flujo de trabajo de pruebas del área temática Aps. (2009)  
<http://www.articuloz.com/informatica-y-tecnologia-articulos/propuesta-de-actividades-del-proceso-de-aseguramiento-de-la-calidad-del-flujo-de-trabajo-de-pruebas-del-area-tematica-aps-886029.html>
- [12] Yaremys López Fojo, Costos de calidad. En busca de la calidad de gestión  
<http://www.monografias.com/trabajos75/costos-calidad-calidad-gestion/costos-calidad-calidad-gestion2.shtml>
- [13] Juan Vázquez y Carlos Doce, Consultores Senior de Seguridad de la Información de TB.  
[http://www.borrmart.es/articulo\\_redseguridad.php?id=2055&numero=39](http://www.borrmart.es/articulo_redseguridad.php?id=2055&numero=39)

[14] Verificación y Validación

<http://www.di.ujaen.es/ asignaturas/computacionestadistica/pdfs/tema6.pdf>

[15] Revista Española de Innovación, Calidad e Ingeniería del Software, volumen 4, No. 4, diciembre, 2008

<http://www.ati.es/IMG/pdf/SanzVol4Num4.pdf>

[16] Enrique Castillo San Martín. La importancia del Testing.

[http://www.economiadehoy.com/modulos/mod\\_periodico/pub/mostrar\\_noticia.php?id=20282&idsec=14](http://www.economiadehoy.com/modulos/mod_periodico/pub/mostrar_noticia.php?id=20282&idsec=14)

[17] Revisiones técnicas formales

<http://www.mitecnologico.com/Main/RevisionesTecnicasFormales>

[18] rfernandez, Outsourcing

<http://www.monografias.com/trabajos10/outso/outso.shtml>

[19] Gelperin, D., Hetzel, B. The Growth of Software Testing. Communications of the ACM, Volume 31, nº 6. (1998).

[20] Turing. A. Computing Machinery and Intelligence. Mind 59 (Oct. 1950)

[21] Baker. C. Review of D.D. McCracken's "Digital Computer Programming". Mathematical Tables and Other Aids to Computation. (Oct. 1957)

[22] Goodenough. J.B y Gerhart. S. L. "Toward a Theory of Test Data Selection". IEEE Transactions on Software Engineering. (Jun 1975)

[23] Howden W.E. Theoretical and Empirical Studies of Program Testing. Proceedings of the 3rd international conference on Software engineering. SIGSOFT. (1978)

[24] Miller. E. F. "Program Testing – An overview for managers. " Computer Software and Applications Conference. The IEEE Computer Society's Second International. (Nov 1978)

[25] Myers. G. J. The Art of Software Testing. John Wiley & Sons. New York 1979.

[26] Howden W.E. "Reliability of the Path Analysis Testing Strategy". IEEE Transactions on Software Testing. (Sept 1976)

[27] Miller. E. y Howden.W.E. Eds. "Software Testing and Validation Techniques". IEEE Computer Society Press. New York 1981.

[28] Guideline for Lifecycle Validation, Verification, and Testing of Computer Software. National Bureau of Standards Report NBS FIPS 101. Washington, D.C. 1983.

[29] ANSI / IEEE STD 829-1983 Standard for Software Test Documentation. Institute of Electrical and Electronics Engineers. New York. 1983.

- [30] ANSI / IEEE STD 1012-1986 Standard for Software Verification and Validation Plans. Institute of Electrical and Electronics Engineers. New York. 1986.
- [31] Hetzel. W. The Complete Guide to Software Testing. Second Edition. QED Information Sciences. Wellesley. Mass. 1988.
- [32] Beizer. B. “SW Testing and Quality Assurance”. Van Nostrand Reinhold. 1984.
- [33] Beizer. B. “Software System Testing Techniques” Second Edition, Van Nostrand Reinhold Company Limited.1990.
- [34] Hartmann. J., Imoberdorf. C. y Meisinger. M. “UML-Based Integration Testing”. Proceedings of the International Symposium on Software Testing and Analysis. ACM SUGSOFT Software Engineering Notes. (Aug 2000)
- [35] (Zahran, 1998) Capítulo 5. Ingeniería del Software.  
[http://www.tdr.cesca.es/TESIS\\_UPC/AVAILABLE/TDX-0716102-102210//05Capitulo05.pdf](http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0716102-102210//05Capitulo05.pdf)
- [36] Hammer y Champy Capítulo 5: Ingeniería del Software,(1993, p. 35)  
[http://www.tdr.cesca.es/TESIS\\_UPC/AVAILABLE/TDX-0716102-102210//05Capitulo05.pdf](http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0716102-102210//05Capitulo05.pdf)
- [37] Ian Sommerville, Ingeniería del Software. Séptima edición. PEARSON EDUCACIÓN, S.A. Madrid, (2005)  
[http://books.google.es/books?id=gQWd49zSut4C&printsec=frontcover&source=gbgsge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.es/books?id=gQWd49zSut4C&printsec=frontcover&source=gbgsge_summary_r&cad=0#v=onepage&q&f=false)
- [38] Javier Tuya Isabel Ramos Román Javier Dolado Cosín. Técnicas cuantitativas para la gestión en la ingeniería del software Ed. Netbiblo (2007)  
<http://books.google.es/books?id=PZQoZ9KTNaEC&printsec=frontcover&dq=Javier+Tuya+Isabel+Ramos+Rom%C3%A1n+Javier+Dolado+Cos%C3%ADn.+T%C3%A9cnicas+cuantitativas+para+la+gesti%C3%B3n+en+la+ingenier%C3%ADa+del+software&hl=es#v=onepage&q&f=false>
- [39] Schaeffer, Mark: DoD Systems Engineering and CMMI. CMMI Technology Conference and UserGroup,  
<http://www.dtic.mil/ndia/2004cmmi/CMMIGS/SchaefferCMMI17Nov04v3.pdf>.
- [40] PYSTER Arthur, What Beyond CMMI Is Needed to Help Assure Program and Project Success? Springer, Berlin, Lecture notes in computer science ISSN 0302-9743
- [41] Software Engineering Institute,  
<http://www.sei.cmu.edu/>

- [42] Erik van Veenendaal, Guidelines for Testing Maturity “The Test Maturity Model”, STEN Journal, Vol. IV, 2006
- [43] Ron Swinkels, A comparison of TMM and other Test Process Improvement Models, Frits Philips Institute
- [44] Andrew Goslin, Klaus Olsen, Fran O’Hara, Mac Miller, Geoff Thompson, Brian Wells, Test Maturity Model Integration (TMMi), TMMI Foundation 2008, Editor: Erik van Veenendaal
- [45] rfernandez, Outsourcing  
<http://www.monografias.com/trabajos10/outso/outso.shtml>
- [46] InfoWare SC, Outsourcing en Verificación y Validación  
<http://www.infowaresc.com/Infoware/index.php/outsourcing>
- [47] Object management group.  
<http://www.omg.org>.
- [48] Juan Diego Pérez, Notaciones y lenguajes de procesos. Una visión global.  
<http://www.lsi.us.es/docs/doctorado/memorias/Perez,%20Juan%20D.pdf>
- [49] M.C. Paulk, C.V. Weber, B. Curtis, M.B. Chrisses, The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, Reading, MA, 1995.
- [50] A. Fuggetta. Software Process: A Roadmap. 22nd International Conference on Software Engineering (ICSE’2000), Future of Software Engineering Track, June 4-11, Limerick (Irlanda), ACM, 2000.
- [51] M.I. Kellner, U. Becker-Kornstaedt, W.E. Riddle, J. Tomal, M. Verlage, Process guides: effective guidance for process participants, Proceedings of the Fifth International Conference on the Software Process, 1998, pp. 11–25.
- [52] M. Kellner and R. Phillips. “Practical Technology for Process Assets,” 107-112. Proc. 8th Intern. Software Process Workshop: State of the Practice in Process Technology. Wadern, Germany, March 2-5, 1993. IEEE Computer Soc. Press, 1993.
- [53] J. W. Armitage et al. Software Process Definition Guide: Content of Enactable Software Process Representations (CMU/SEI-94-SR-21) Pittsburgh: PA: Software Engineering Inst., Carnegie Mellon Univ., December 1994.
- [54] Proyecto Fin de Carrera de Ángel Del Saz Sánchez: Desarrollo de una Guía Electrónica de Procesos de Ingeniería de Software. (2009)
- [55] Zonnect. Zonnect Coach (Guía de procesos)  
[http://www.zonnect.com/plataforma/zonnect\\_coach.aspx](http://www.zonnect.com/plataforma/zonnect_coach.aspx)



# Anexos

## ANEXO 1

# CHECKLIST PARA VERIFICAR CÓDIGO FUENTE

## ANEXO 1: CHECKLIST PARA VERIFICAR CÓDIGO FUENTE

En este anexo se encuentra la Checklist necesaria para poder verificar que el código fuente de cualquier producto software es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá argumentar una explicación que refleje el motivo.

Verificación del código fuente			
CUESTIONES DE DISEÑO			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Cada unidad de ejecución tiene una única función?			
¿Hay casos donde la unidad tenga particiones?			
¿El código es coherente con el diseño detallado?			
¿Cubre el código el diseño detallado?			
ELEMENTOS DE DATOS			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Ha sido comprobada la validez de la entrada?			
¿Han sido definidas e inicializadas todas las variables?			
¿El tipo de las variables es correcto?			
¿Son usadas todas las variables?			
¿Se ha comprobado el alcance de las variables?			
CÁLCULOS			

PREGUNTA	SI	NO	EXPLICACIÓN
¿En los cálculos se ha utilizado variables cuyo tipo de datos es inconsistente?			
¿Hay cálculos de modo mixto?			
¿Es válido usar número enteros?			
¿Es válido usar aritmética de punto flotante?			
¿Hay comparaciones entre los números de coma flotante?			
¿Existen suposiciones sobre el orden de evaluación en las expresiones booleanas?			
¿Los operadores de comparación son correctos?			
<b>CUESTIONES SOBRE EL FLUJO DE CONTROL</b>			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Terminará eventualmente el programa, módulo o unidad?			
¿Hay posibilidad de bucle infinito?			
¿Hay posibilidad de un bucle con salida prematura?			
¿Hay posibilidad de que haya un bucle que nunca se ejecute?			
<b>CUESTIONES DE INTERAFACES</b>			

PREGUNTA	SI	NO	EXPLICACIÓN
¿El número de parámetros utilizados coinciden con los de la rutina de la llamada?			
¿Es el orden de los parámetros correcto y consistente en las llamadas y en el destinatario?			
¿Hay una función que tenga solo un parámetro de entrada?			
¿Hay un procedimiento que tenga solo un parámetro de entrada?			
¿Hay variables globales?			
¿Las variables globales tienen sus definiciones en todos los módulos donde se usan?			
¿Las variables globales tienen sus atributos en todos los módulos donde se usan?			
<b>CUESTIONES DE ENTRADA/SALIDA</b>			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Se han abierto todos los archivos de uso?			
¿Los ficheros han sido correctamente cerrados cuando se ha terminado?			
¿Han sido declarados todos los ficheros?			
¿Son correctos los atributos de los ficheros?			
¿Los errores en EOF se han entregado correctamente?			

¿Los errores en I/O se han entregado correctamente?			
¿Es compatible el tamaño del buffer de I/O con el tamaño del registro?			
<b>CUESTIONES DE PORTABILIDAD</b>			
<b>PREGUNTA</b>	<b>SI</b>	<b>NO</b>	<b>EXPLICACIÓN</b>
¿Hay un conjunto de caracteres asumido?			
¿Hay asumido números enteros o representación de punto flotante?			
¿Es necesario modificadas las llamadas de su servicio?			
<b>MENSAJES DE ERROR</b>			
<b>PREGUNTA</b>	<b>SI</b>	<b>NO</b>	<b>EXPLICACIÓN</b>
¿Se han usado adecuadamente todos los warnings y mensajes de información?			
<b>DOCUMENTACIÓN DE CÓDIGO Y COMENTARIOS</b>			
<b>PREGUNTA</b>	<b>SI</b>	<b>NO</b>	<b>EXPLICACIÓN</b>
¿Está el código documentado correctamente?			
¿En el código hay procedimientos apropiados?			
¿Los comentarios en el código son adecuados?			
¿Es una documentación clara y correcta?			

DISPOSICIÓN DEL CÓDIGO Y ESPACIOS EN BLANCO			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Se ha utilizado sangría y espacios en blanco para apoyar a la comprensión del código?			
MANTENIMIENTO			
PREGUNTA	SI	NO	EXPLICACIÓN
¿Cada modulo tiene solo un punto de salida?			
¿Son módulos fáciles de cambiar?			

Tabla 6: Checklist verificar código fuente

## ANEXO 2

### CHECKLIST PARA LAS ACTIVIDADES DEL PRODUCTO DE TRABAJO



## ANEXO 2: CHECKLIST PARA LAS ACTIVIDADES DEL PRODUCTO DE TRABAJO

### Producto de trabajo: DOCUMENTO DE REQUISITOS

En este anexo se encuentra la Checklist necesaria para poder verificar que las actividades del producto de trabajo “documento de requisitos” es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha determinado el alcance del sistema?			
¿Se ha realizado un estudio de la situación actual?			
¿Han sido identificados los proveedores encargados de extraer los requisitos de usuario?			
¿Se han determinado los requisitos de usuario?			
¿Han sido registrados los requisitos de usuario?			
¿Se han definido los requisitos software?			
¿Han sido definidos los requisitos de seguridad y operación?			
¿Se ha elaborado la propuesta de sistema de información?			
¿Ha sido revisado la propuesta de sistema de información?			
¿Ha sido aprobado la propuesta de sistema de información?			

Tabla 7: Checklist documento de requisitos

**Producto de trabajo: ANÁLISIS DEL SISTEMA**

**Análisis estructurado:**

En este anexo se encuentra la Checklist necesaria para poder verificar que las actividades del producto de trabajo “Análisis del Sistema” para Análisis estructurado es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha realizado una definición del sistema que indique que hará el sistema?			
¿Con qué se hará el sistema?			
¿Quién se comunicará con el sistema?			
¿Quién usará el sistema?			
¿Qué estándares son aplicables al sistema?			
¿Se ha realizado un estudio de la seguridad requerida en el análisis del sistema?			
¿Se ha especificado el diagrama de flujos de datos?			
¿Han sido determinados los subsistemas de análisis?			
¿Ha sido realizado el modelo de procesos del sistema?			
¿Se ha realizado el modelo conceptual de datos?			
¿Se ha realizado el modelo lógico de datos?			
¿Han sido especificados los principios generales de la interfaz?			
¿Han sido establecidos los diferentes perfiles de usuario?			
¿Se ha especificado los formatos individuales de la interfaz de pantalla?			
¿Se han especificado los formatos de impresión?			
¿Han sido generados el documento de análisis para revisión?			
¿Se ha realizado la revisión del análisis funcional?			

¿Se ha obtenido la aprobación del análisis del sistema de información?			
--	--	--	--

**Tabla 8: Checklist análisis estructurado**

### **Análisis orientado a objetos**

En este anexo se encuentra la Checklist necesaria para poder verificar que las actividades del producto de trabajo “Análisis del Sistema “para Análisis orientado a objetos es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha realizado una definición del sistema que indique que hará el sistema?			
¿Con qué se hará el sistema?			
¿Quién se comunicará con el sistema?			
¿Quién usará el sistema?			
¿Qué estándares son aplicables al sistema?			
¿Han sido especificados los casos de uso?			
¿Se ha realizado un estudio de la seguridad requerida en el análisis del sistema?			
¿Han sido determinados los subsistemas de análisis?			
¿Se han identificado las clases asociadas a un caso de uso?			
¿Ha sido descrita la interacción de los objetos en base a cada uno de los casos de uso?			
¿Se han especificado los principios generales de la interfaz?			
¿Han sido identificados un catalogo de usuarios que facilite el análisis de las interfaces de usuarios del sistema?			
¿Se han especificado los elementos estáticos de la interfaz de pantalla?			
¿Se han especificado los formatos de impresión?			
¿Se ha identificado las responsabilidades y atributos de las clases?			

¿Se ha identificado las asociaciones y agregaciones?			
¿Se ha identificado las relaciones de herencia entre las clases?			
¿Se ha desarrollado el diagrama de clases?			
¿Han sido generados el documento de análisis para revisión?			
¿Se ha realizado la revisión del análisis funcional?			
¿Se ha obtenido la aprobación del análisis del sistema de información?			

**Tabla 9: Checklist análisis orientado a objetos**

## **Producto de trabajo: DISEÑO DEL SISTEMA**

### **Diseño estructurado**

En este anexo se encuentra la Checklist necesaria para poder verificar que las actividades del producto de trabajo “Diseño del Sistema “para el Diseño estructurado es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se han definido las normas de grabación y diseño de registros?			
¿Se han identificado los subsistemas de diseño?			
¿Han sido diseñados los módulos del sistema?			
¿Se ha diseñado las comunicaciones entre módulos?			
¿Han sido generadas las especificaciones necesarias para la construcción del sistema de información?			
¿Se han especificado las necesidades de migración de datos y cómo se realizaría la carga inicial de los datos implicados?			
¿Se ha generado el documento de diseño para revisión?			
¿Se ha realizado la revisión del diseño funcional?			
¿Ha sido aprobado el diseño del sistema de información?			
¿Ha sido elaborado el plan de pruebas del sistema?			

¿Ha sido elaborado el plan de pruebas de aceptación?			
¿Ha sido definido el entorno necesario para la realización de las pruebas de sistema?			
¿Ha sido definido el entorno necesario para la realización de las pruebas de aceptación?			
¿Han sido definidas y organizadas las pruebas del sistema?			
¿Han sido definidas y organizadas las pruebas de aceptación?			

**Tabla 10: Checklist diseño estructurado**

### **Diseño orientado a objetos**

En este anexo se encuentra la Checklist necesaria para poder verificar que las actividades del producto de trabajo “Diseño del Sistema “para el Diseño orientado a objetos es correcto. Para ello se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se han definido las normas de grabación y diseño de registros?			
¿Se han definido los componentes del sistema?			
¿Ha sido especificado el entorno tecnológico?			
¿Ha sido diseñado el comportamiento de componentes y clases?			
¿Se han generado las especificaciones necesarias para la construcción del sistema de información?			
¿Se han especificado las necesidades de migración de datos y cómo se realizaría la carga inicial de los datos implicados?			
¿Se ha generado el documento de diseño para revisión?			
¿Se ha realizado la revisión del diseño funcional?			
¿Ha sido aprobado el diseño del sistema de información?			
¿Ha sido elaborado el plan de pruebas del sistema?			
¿Ha sido elaborado el plan de pruebas de aceptación?			

¿Ha sido definido el entorno necesario para la realización de las pruebas de sistema?			
¿Ha sido definido el entorno necesario para la realización de las pruebas de aceptación?			
¿Han sido definidas y organizadas las pruebas del sistema?			
¿Han sido definidas y organizadas las pruebas de aceptación?			

**Tabla 11: Checklist diseño orientado a objetos**

## ANEXO 3

### CHECKLIST PARA LOS PRODUCTOS DE TRABAJO

### **ANEXO 3: CHECKLIST PARA LOS PRODUCTO DE TRABAJO**

En este anexo se encuentra la Checklist necesaria para poder verificar que los productos de trabajo obtenidos son correctos.

Para garantizar que el producto de trabajo “Documento de Requisitos “es correcto se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

Producto de Trabajo: Documento de Requisitos			
COMPLETITUD E INTEGRIDAD			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Han sido completados todos los elementos esenciales?			
¿Han sido omitidos todos los elementos irrelevantes?			
¿El nivel técnico de cada tema está abordado adecuadamente para este documento?			
¿Hay una declaración clara de los objetivos de este documento?			
¿Son los objetivos coherentes con la política?			
¿Puede ser eliminado algún requisito?			
¿Pueden juntarse dos requisitos?			
¿Se han especificado todos los recursos hardware necesario?			
¿Se han especificado las interfaces externas necesarias?			
¿Han sido definidos los criterios de aceptación para cada una de las funciones especificadas?			
EXACTITUD			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Hay algún elemento incorrecto?			
¿Hay alguna contradicción?			
¿Hay alguna ambigüedad?			



CLARIDAD Y CONSISTENCIA			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Son los materiales y las declaraciones claros en el documento?			
¿Son los ejemplos claros, útiles, relevantes y correctos?			
¿Son los diagramas, gráficos e ilustraciones claras y efectivos?			
¿Se usan en los diagramas, gráficos e ilustraciones una notación propia y se encuentran en el lugar adecuado?			
¿La terminología es clara y correcta?			
¿Hay un glosario de términos técnicos completo y correcto?			
¿El estilo de escritura es claro?			
REFERENCIAS Y AYUDAS PARA DOCUMENTAR LA COMPRENSIÓN			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se dispone de un resumen o introducción?			
¿Hay una buena tabla de contenidos?			
¿Son los elementos rotos fáciles de seguir y comprensibles?			
¿Hay una bibliografía clara, completa y correcta?			
¿Hay un índice claro, completo y correcto?			
¿Es la página y la numeración de la figura correcta y consistente?			

Tabla 12: Checklist documento de requisitos

Para garantizar que el producto de trabajo “Análisis del Sistema” es correcto se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

Producto de Trabajo: Análisis del Sistema			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha dado un nombre al proyecto?			
¿Han sido identificadas todas las necesidades del cliente?			
¿Han sido identificadas las metas globales del proyecto?			
¿Han sido identificadas todas las tareas?			
¿Se ha especificado si es viable económicamente el sistema?			
¿Se ha especificado si es viable técnicamente el sistema?			
¿Se ha realizado un análisis económico?			
¿Se ha realizado un análisis técnico?			
¿Se han definido las restricciones de tiempo y presupuesto?			
¿Se han asignado funciones al Hardware?			
¿Se han asignado funciones al software?			
¿Se han identificado los participantes del proyecto?			
¿Se han asignado funciones al personal?			
¿Se han asignado funciones a la base de datos?			
¿Se ha realizado una evaluación de los enfoques alternativos del sistema?			
¿Se ha documentado el modelo de la arquitectura del sistema?			

Tabla 13: Checklist Análisis del Sistema

Para garantizar que el producto de trabajo “Diseño del Sistema” es correcto se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

Producto de Trabajo: Diseño del Sistema			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Cubre el diseño todos los requisitos funcionales?			
¿Resulta ambigua la documentación del diseño?			
¿Es la técnica de diseño utilizada la mejor opción?			
¿La notación usada es correcta?			
¿La solución elegida es la más simple?			
¿Es un diseño suficientemente detallado como para poder implementarlo en el lenguaje de programación elegido?			
¿Se ha determinado la calidad del modelo de arquitectura de alto nivel?			
¿Se ha descrito el modulo de las interfaces?			
¿Las interfaces de usuario son claras?			
¿Las ayudas ofrecidas al usuario son de calidad?			
¿Se han identificado los criterios de ejecución y las secuencias operativas?			
¿Hay una descripción clara entre las interfaces de este sistema y otro sistema software y hardware?			
¿Son completos todos los requisitos funcionales de los elementos de diseño?			
¿Se ha revisado la ocultación de información?			
¿Se ha revisado la herencia?			
¿Se ha revisado la cohesión del modulo y el enganche?			
¿Hay calidad en la descripción de las interfaces de módulos?			
¿Se ha reutilizado módulos?			

**Tabla 14: Checklist Diseño del Sistema**

Para garantizar que el producto de trabajo “Plan del Proyecto “es correcto se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

Producto de Trabajo: Plan del Proyecto			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha elegido el ciclo de vida adecuado?			
¿Se ha realizado una correcta definición del objetivo del proyecto?			
¿Es la prueba de costes compatible con las especificaciones del plan de proyecto?			
¿Se han realizado cambios en los requerimientos?			
¿Se ha realizado un balance entre el tiempo, costo, funcionalidades, calidad y recursos frente a las expectativas del cliente?			
¿Se han definido todas las variables y marcado cuales son los puntos de los requerimientos que no serán alcanzados?			
¿Se han asignado bien los recursos?			
¿Se ha definido el cronograma correctamente?			
¿Se ha definido todos los procedimientos necesarios?			
¿Se ha definido el entorno donde se desarrollará el proyecto?			
¿Se ha definido cuales son los cambios permitidos que afectan al requerimiento o dominio del problema?			
¿Se han programado la comunicación necesaria para realizar el proyecto?			
¿Se ha establecido un compromiso?			
¿Se han identificado los beneficios del proyecto?			
¿Coinciden con la realidad dichos beneficios?			
¿Se han analizado los riesgos del proyecto?			

**Tabla 15: Checklist Plan de Proyecto**

Para garantizar que el producto de trabajo “Plan de Pruebas del Proyecto” es correcto se deberá de formular las diferentes cuestiones que contiene la Checklist y anotar para cada una si se ha llevado a cabo o no. En caso de que la respuesta sea negativa se deberá justificar el motivo.

## Producto de Trabajo: Plan de Pruebas del Proyecto

### **PRUEBA DE ELEMENTOS**

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se encuentran todos los elementos que deben ser probados en el Plan de pruebas?			
¿Han sido cubiertas en el plan de pruebas todos los requerimientos, características y elementos de diseño?			
¿Ha sido descrito claramente la manera de enfocar la prueba?			
¿Ha sido descrito claramente el criterio de fallo/ acierto?			
¿Ha sido descrito claramente el criterio de suspensión y reinicio?			
¿Se han incluido los resultados de todas las pruebas?			

### **PERSONAL DE PROGRAMACIÓN Y RESPONSABILIDADES**

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Han sido todas las tareas de pruebas definidas, asignadas y programadas?			
¿Son compatibles las responsabilidades y programaciones con el proyecto programado en general?			
¿Han sido previstas y dirigidas en la programación las necesidades de entrenamiento o formación?			

### **PRUEBAS DEL MEDIO AMBIENTE**

PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se han definido las pruebas de necesidades de software y hardware?			
¿Ha sido asignado el tiempo para la creación y destrucción del medio?			
¿Han sido identificados y analizados todos los riesgos asociados a la prueba de los productos de software?			

<b>PRUEBA DE COSTES</b>			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Se ha definido un plan de cuentas para las pruebas de costes?			
¿Es la prueba de costes compatible con las especificaciones del plan de proyecto?			
<b>ACCESORIOS DEL PLAN DE PRUEBAS</b>			
PREGUNTA	SI	NO	JUSTIFICACIÓN
¿Han sido completadas todas las especificaciones de diseño probadas?			
¿Se ha desarrollado una matriz de trazabilidad para asegurar todos los requerimientos, características y elementos de diseño cubiertos en las pruebas?			
¿Han sido especificados los casos probados?			
¿Son correctos los casos probados?			
¿Son completos los casos probados?			
¿Han sido establecidos los vínculos adecuados entre los casos de pruebas procedimientos y diseños?			
¿Son las pruebas adecuadas a cada elemento?			
¿En el test de integración ha sido incluido, explicado y documentado el enfoque de integración?			
¿Han sido documentados los procedimientos de las pruebas?			
¿Los procedimientos de las pruebas son completos y de acuerdo a las normativas?			
¿Ha sido especificada la documentación requerida tras la prueba?			
¿Se ha dado oportunidad a los usuarios de participar en los tests de aceptación?			

**Tabla 16: Checklist Plan de Pruebas del Proyecto**

## ANEXO 4

# ESPECIFICACIÓN DEL PROCESO ASEGURAMIENTO DE LA CALIDAD

## **ANEXO 4: ESPECIFICACIÓN DEL PROCESO ASEGURAMIENTO DE LA CALIDAD**

### ESPECIFICACIÓN DEL PROCESO ASEGURAMIENTO DE LA CALIDAD

#### ÍNDICE DE CONTENIDOS:

1. LAS ACTIVIDADES DE CALIDAD A REALIZAR 3
2. PLANIFICAR LAS ACTIVIDADES DE CALIDAD 4
3. REALIZAR LAS ACTIVIDADES DE CALIDAD 5
4. DOCUMENTAR Y RESOLVER LAS INCIDENCIAS 6
5. DOCUMENTAR Y RESOLVER LAS LECCIONES APRENDIDAS 7



## **ASEGURAMIENTO DE LA CALIDAD**

### **Visión General**

El objetivo principal del proceso de Aseguramiento de la Calidad es llevar a cabo el seguimiento de las actividades de calidad comprometidas internamente en el servicio. Es decir, ejecutar aquellas acciones que hacen que un proceso cumpla con unos determinados requisitos de calidad.

Para ello, será necesario verificar que cada una de las actividades, que se han definido en los procesos implantados en la organización, se está llevando a cabo y, además, se está realizando correctamente, es decir se obtienen los productos intermedios o finales adecuados como resultado de esa actividad. Por tanto, se verificará que las actividades y productos resultantes se realizan conforme a lo establecido en los procesos, instrucciones técnicas, o cualquier activo de proceso especificado en los distintos procesos.

### **Criterios de Entrada**

- **Productos de trabajo**  
Conjunto de los productos de trabajo obtenidos como resultado de la realización de las actividades.
- **Documentación de los procesos**  
Documentación de los procesos implantados en la organización.
- **Políticas de aseguramiento de la calidad**  
Políticas de aseguramiento de la calidad que se encuentren vigentes en la organización.
- **Procedimientos**  
Procedimientos vigentes en la organización para la realización de auditorías.
- **Documentos de procesos**  
Documentos de procesos a evaluar.

**Cuándo:** Una vez se haya iniciado la ejecución del servicio.

## **1. LAS ACTIVIDADES DE CALIDAD A REALIZAR**

### **Visión General**

El objetivo principal de esta actividad es identificar y determinar el conjunto de actividades de Aseguramiento de la Calidad a realizar.

Para poder determinar las actividades de calidad será necesario previamente seleccionar el conjunto de procesos que se van a considerar como aplicables. Una vez se haya realizado esta selección, se podrán determinar los mecanismos que van a ser utilizados para verificar su correcta aplicación en el ámbito del servicio.

Cada uno de los procesos tendrá asociado un conjunto de ítems que permiten verificar su correcta aplicación en un servicio. Estos ítems se denominan elementos de verificación. Así, a partir de los procesos seleccionados para el proyecto, de las Listas de Verificación generales asociados a los mismos (incluyendo los ítems de verificación correspondientes) y de la información incluida en el catálogo del servicio, se determinarán los elementos de verificación que se deben controlar para asegurar la calidad del servicio en curso. Asimismo, también se deberá indicar que elementos de verificación se excluyen de las actividades de calidad configurando las listas de verificación de cada proceso aplicable en un proyecto determinado.

**Cuándo:** Una vez se haya iniciado la ejecución del servicio.

### **Roles Participantes**

#### **Responsables**

- Ingeniero de pruebas
- Responsable del Servicio

#### **Criterios de Entrada**

- **Documentación de los procesos**  
Documentación de los procesos implantados en la organización.
- **Políticas de aseguramiento de la calidad**  
Políticas de aseguramiento de la calidad que se encuentren vigentes en la organización.
- **Catalogo de servicios**  
Descripción sobre los servicios disponibles.

## **TAREAS**

**Identificar procesos aplicables:** Se identifican el conjunto de procesos aplicables en función de qué servicios se desean evaluar, cuáles de ellos están debidamente documentados y disponibles en la librería de activos y, la lista general de verificación.

**Determinar actividades de calidad:** A partir de la lista general de verificación y los procesos aplicables, se proponen las actividades de calidad a aplicar.

**Acordar las actividades de calidad:** Se realiza un acuerdo sobre las actividades identificadas.

### **Criterios de Salida**

- **Lista de verificación**

Lista de verificación que contiene las actividades aplicables que han sido acordadas.

### **Elementos de Trabajo**

Elemento de verificación

### **Productos de Trabajo**

No tiene

### **Instrucciones técnicas**

Listas de verificación para los servicios existentes

## **2. PLANIFICAR LAS ACTIVIDADES DE CALIDAD**

### **Visión General**

El objetivo principal de esta actividad es realizar la planificación de las actividades de calidad que han sido previamente acordadas.

Con el propósito de realizar las actividades de aseguramiento de calidad de una forma ordenada y controlada será necesario planificarlas, teniendo en cuenta la ejecución de los servicios afectados. Se realizará la asignación de recursos y de responsables y se elaborará el calendario de realización de las mismas.

Básicamente la realización de cada una de estas actividades consistirá en cumplimentar adecuadamente las listas de verificación acordadas en función de los hallazgos encontrados.

### **Roles Participantes**

#### **Responsables**

- Ingeniero de Pruebas

- Responsable del Servicio

**Cuándo:** Se han identificado las actividades a realizar.

#### **Criterios de Entrada**

- **Lista de verificación**

Lista de verificación que contiene las actividades aplicables que han sido acordadas.

**Asignar recursos:** Identificar y asignar el conjunto de recursos que van a desarrollar las actividades de calidad.

**Determinar calendario:** Identificar y establecer las fechas durante las cuales se van a desarrollar las actividades de calidad.

**Asignar responsables:** Asignar a cada actividad la persona responsable de su desarrollo.

#### **Criterios de Salida**

- **Planificación de las actividades de calidad**

#### **Elementos de Trabajo**

Esta actividad no tiene elementos de trabajo asociados.

#### **Productos de Trabajo**

No tiene.

#### **Instrucciones técnicas**

No tiene.

### **3. REALIZAR LAS ACTIVIDADES DE CALIDAD**

#### **Visión General**

El objetivo principal de esta actividad es llevar a cabo cada una de las actividades de calidad que han sido planificadas previamente.

Los responsables que hayan sido asignados para realizar las actividades de calidad, las llevarán a cabo en las fechas previstas y de acuerdo con las normas y técnicas de trabajo acordadas.

Para ello, cada uno de responsable deberá cumplimentar las listas de verificación a medida que dichas actividades se vayan realizando o justificando debidamente aquellas actividades comprometidas y que no se hayan realizado.

**Cuándo:** Se han planificado las actividades a realizar

### **Roles Participantes**

#### **Responsables**

- Ingeniero de pruebas
- Responsable del servicio

**Cuando:** Se hayan planificado las actividades de calidad.

#### **Criterios de Entrada**

- **Lista de verificación**  
Lista de verificación que contiene las actividades aplicables que han sido acordadas.
- **Planificación de las actividades de calidad**
- **Documentación del servicio**  
Incluye la documentación que existente relacionada con el servicio.
- **Criterios, normas y procedimientos**  
Criterios, normas y procedimientos que sean necesarios para realizar las actividades de calidad.

**Realizar las actividades de calidad:** Recoger y analizar la información necesaria para cumplimentar las listas de verificación.

**Cumplimentar lista de verificación:** A partir de los datos obtenidos del análisis de las actividades realizadas y los productos obtenidos, se cumplimenta la lista de verificación.

#### **Criterios de Salida**

- **Lista de verificación**  
Lista de verificación cumplimentada.
- **Elementos de Trabajo**  
Esta actividad no tiene elementos de trabajo asociados.

#### **Productos de Trabajo**

Se podrían poner las Checklist de los procesos.

#### **Instrucciones Técnicas**

No tiene.

## 4. DOCUMENTAR Y RESOLVER LAS INCIDENCIAS

### Visión General

El objetivo principal de esta actividad es documentar y resolver las incidencias que surjan durante el desarrollo del servicio como consecuencia de que o bien no se ha realizado una actividad de un proceso definido o bien porque ha sido realizada con métodos distintos a los especificados en dichos procesos.

Las incidencias identificadas durante el seguimiento de las actividades de calidad serán cumplimentadas describiendo cómo, quién y cuándo deben ser resueltas. Además, las acciones recomendadas deberán ser seguidas y controladas, realizándose el su cierre de las mismas cuando hayan sido realizadas correctamente, indicando la fecha de cierre y dejando constancia de dicho cierre.

**Cuándo:** Durante la realización de las actividades de calidad.

### Roles Participantes

#### Responsables

- Ingeniero de Pruebas
- Responsable del Servicio

#### Informados

- Responsable de la unidad organizativa

### Criterios de Entrada

- **Lista de verificación**  
Lista de verificación cumplimentada.

**Identificar incidencias:** Analizar la lista de verificación que ha sido cumplimentada para identificar las incidencias.

**Documentar incidencias:** Documentar cada una de las incidencias identificadas, incluyendo cómo resolverlas, quién y en qué momento.

**Resolver las incidencias:** Llevar a cabo las acciones recomendadas para la resolución de las incidencias.

**Informar de las incidencias:** Una vez realizada la documentación y resolución de las incidencias encontradas se procede a informar al responsable de la unidad organizativa.

### **Criterios de Salida**

- Informe de Incidencias documentadas y cerradas

### **Elementos de Trabajo**

Incidencia, Elemento de Verificación.

### **Productos de Trabajo**

Informe de Incidencias.

### **Instrucciones técnicas**

Esta actividad no tiene instrucciones técnicas.

## **5. DOCUMENTAR Y RESOLVER LAS LECCIONES APRENDIDAS**

### **Visión General**

El objetivo principal de esta actividad es documentar y resolver las lecciones aprendidas que se hayan obtenido durante la ejecución del servicio.

Las lecciones aprendidas serán las prácticas excelentes y las prácticas deficientes que han sido realizadas durante el desarrollo del proyecto. Éstas podrán ser extraídas a partir de incidencias que han sido identificadas y documentadas, mediante la realización de dinámicas de grupo.

Para ello, al finalizar el proyecto, se realizará una reunión con el equipo de proyecto en la que, para cada área de proceso, se deberán identificar:

- Prácticas Excelentes
- Prácticas Deficientes
- Riesgos Altos

En esta reunión, además del equipo de trabajo y el responsable del servicio, puede participar el Responsable de la Unidad Organizativa.

**Cuando:** Después de documentar y resolver las incidencias del proyecto.

### **Roles Participantes**

#### **Responsables**

- Responsable de la unidad organizativa
- Responsable del servicio

### **Criterios de Entrada**

- Informe de Incidencias documentadas y cerradas
- Lista de verificación
- Lista de verificación cumplimentada

**Identificar las lecciones aprendidas:** Identificar, a partir de las incidencias, cuáles han sido las prácticas excelentes, las prácticas deficientes y cuáles son los riesgos altos.

**Documentar las lecciones aprendidas:** Documentar las lecciones aprendidas, detallando el área de proceso al que referencian y las personas involucradas.

### **Criterios de Salida**

- Lecciones aprendidas documentadas

### **Elementos de Trabajo**

Esta actividad no tiene elementos de trabajo asociados.

### **Productos de Trabajo**

No tiene.

### **Guías**

Dinámica para la Generación de Lecciones Aprendidas.



## ANEXO 5

### Plan de Aseguramiento de la Calidad

## **ANEXO 5: PLAN DE ASEGURAMIENTO DE LA CALIDAD**

### **PLAN DE ASEGURAMIENTO DE CALIDAD**

#### **ÍNDICE DE CONTENIDOS**

- PROPÓSITO
- REFERENCIAS
- ROLES Y RESPONSABILIDADES
- DOCUMENTACIÓN
- ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS
- REVISIONES Y AUDITORIAS
- REPORTE DE PROBLEMAS
- TÉCNICAS, HERRAMIENTAS Y METODOLOGÍAS
- MECANISMOS DE CONTROL

## 1. PROPÓSITO

El plan de aseguramiento de calidad busca establecer los responsables, fases, herramientas, técnicas, indicadores y documentación que se usarán para asegurar la calidad del producto.

## 2. REFERENCIAS

Hay que especificar las referencias de:

- Proyecto
- Documentación
- Estándares
- Información adicional

## 3. ROLES Y RESPONSABILIDADES

Se trata de identificar las personas responsables del aseguramiento de la calidad indicando nombres, el rol que tienen y las responsabilidades que tienen en el proyecto con relación a este aspecto.

## 4. DOCUMENTACIÓN

Especificar de manera clara y objetiva toda la información que se generará y que se usará en cada fase del proyecto. Se debe desglosar por fases.

## 5. ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS

- Especificar los estándares que se usarán de acuerdo al dominio del conocimiento al que pertenece el proyecto.
- Definir las métricas y cómo se obtendrán.
- Especificar indicadores junto con sus parámetros de interpretación y las métricas que se usarán para su obtención.

## 6. REVISIONES Y AUDITORÍAS

Indicar en qué momentos y qué elementos se revisarán durante el desarrollo del proyecto.

## 7. REPORTE DE PROBLEMAS

Especificar que se reportará y a quién se reportará cuando se encuentren problemas de calidad, además se debe especificar el mecanismo de resolución de conflictos.

## 8. TÉCNICAS, HERRAMIENTAS Y METODOLOGÍAS

- Indicar la metodología que se usará.
- Especificar qué herramientas se usarán junto con las técnicas que apoyen el cumplimiento de la metodología de trabajo.

## 9. MECANISMOS DE CONTROL

Indicar qué mecanismos se usarán para asegurar que cada etapa se cumple conforme lo previsto; ¿dónde se tendrá acceso a la información? y ¿cómo será este acceso?

Figura 68: Producto de Trabajo: Plan de Aseguramiento de la Calidad

## ANEXO 6

### Plan de Gestión de Configuración

## **ANEXO 6: PLAN DE GESTIÓN DE CONFIGURACIÓN**

<p>PLAN DE GESTIÓN DE CONFIGURACIÓN</p>	<p><b>ÍNDICE DE CONTENIDOS</b></p> <p>1. Introducción .....3</p> <p>1.1 Propósito .....3</p> <p>1.2 Alcance .....4</p> <p>1.3 Terminología .....4</p> <p>1.4 Referencias .....4</p> <p>2. Gestión del plan .....5</p> <p>2.1 Organización .....5</p> <p>2.2 Responsabilidades .....5</p> <p>3. Actividades .....6</p> <p>4. Calendario .....6</p> <p>4.1 Calendario de actividades y tareas .....6</p> <p>4.2 Hitos de ejecución .....6</p> <p>5. Mantenimiento del plan .....6</p>
---	---

## 1. Introducción

Para realizar la GC de nuestro PFC realizaremos las siguientes actividades mediante las cuales se garantizará que nuestro PFC va a estar gestionado adecuadamente y de manera íntegra desde su comienzo. La GC del PFC nos va a permitir evaluar y controlar los cambios que hagamos sobre el PFC además de comprobar que el resultado del PFC sea el esperado.

Las actividades para realizar la GC son: identificar los EC, control de cambios, control de versiones y por último realizar una auditoría de la configuración.

### 1.1 Propósito

Durante el proceso de desarrollo de nuestro PFC llevaremos a cabo las siguientes actividades de gestión de configuración. Las actividades para realizar la GC son:

- **Identificar los EC:** Esta actividad tiene como objetivo identificar que productos de trabajo del PFC van a someterse a la GC. Serán aquellos productos de trabajo que son necesarios de controlar en el proyecto.
- **Control de cambios:** Esta actividad tiene como objetivo controlar todos los cambios que se produzcan en los EC seleccionados anteriormente.
- **Control de versiones:** Esta actividad tiene como objetivo almacenar la nueva versión que surja tras realizar un cambio sobre un EC del proyecto.
- **Auditoría de la configuración:** Esta actividad tiene como objetivo evaluar si un EC de nuestro proyecto satisface las expectativas esperadas además de comprobar si un cambio realizado ha sido llevado a cabo correctamente.

A continuación definiremos los productos de nuestro proyecto que se pondrán bajo control de configuración:

- Aseguramiento de la calidad
- Gestión de configuración
- Verificación de productos de trabajo
- Verificación del código fuente
- Pruebas de aceptación y sistemas

## 1.2 Alcance

En este apartado se describe de forma correcta, clara y resumida el alcance real de GC, una buena forma de hacerlo es describir todo lo que va a abordar la GC y lo que no va a abordar.

El alcance de GC aquí descrito comprende:

- Descripción general del proyecto de desarrollo software.
- Identificación de los EC a los que se aplica la GC.
- Software de apoyo o de pruebas.
- Limitaciones del plan: restricciones de tiempo, coste, calendario o capacidad para realizar las actividades de GC.

## 1.3 Terminología

En este apartado se describen todas las definiciones que son necesarias para seguimiento y comprensión del plan o que sirven de ayuda para su comunicación.

- QA: (Quality Assurance) Aseguramiento de calidad
- EC: Elementos de configuración
- CM: (Configuration Management) GC: Gestión de configuración
- LB: Línea base

## 1.4 Referencias

[1] SEL: Software Engineering Lab

[2] Análisis y diseño de aplicaciones informáticas de gestión, Mario G. Piattini, RA-MA

[3] Internet

[4] IEEE Standard for Software Configuration Management Plans



## 2. Gestión del plan

Se describe las responsabilidades y responsables para la realización de las actividades de GC dentro del proyecto.

### 2.1 Organización

Se debe especificar las estructuras organizacionales tanto técnicas como de gestión de proyectos, las cuales participarán en la implementación de actividades de GC.

Se debe identificar:

- Todas las áreas que participen o sean responsables de actividades de GC.
- El cometido de estas áreas dentro del proyecto.
- Relaciones entre estas áreas.

### 2.2 Responsabilidades

En este apartado se describen la asignación de responsabilidades y autoridades para las actividades de GC además de los recursos que van a ser usados en la GC.

El plan deberá explicar para cada actividad de GC las herramientas, técnicas, equipamiento, personas que se necesitan y cómo se proporciona o se obtiene cada recurso.

Actividad	Herramientas	Técnicas	Personal	Equipamiento	Cómo
<Actividad 1>					
<Actividad 2>					
<Actividad n>					

### 3.Actividades

Identifica todas las actividades y tareas que se requieren para GC. Estas deben ser tanto actividades técnicas como de GC, así como las actividades generales del proyecto estén implicadas en la configuración.

Actividades de GC:

<Actividad 1>

<Actividad 2>

<Actividad n>

### 4.Calendario

#### 4.1 Calendario de actividades y tareas

En este apartado se describe la secuencia y coordinación para las actividades de GC. Se define la secuencia y dependencia que existe entre las actividades además de especificar la duración de cada una de ellas y por tanto la duración del Plan.

#### 4.2 Hitos de ejecución

En este apartado se describen todos los hitos del proyecto relativos a actividades de GC y estos hitos incluyen:

- Definición de la línea base.
- Implementación de control de cambios.
- Fechas de comienzo y fin de auditorías.

### 5.Mantenimiento del plan

Esta sección debe contener:

- Quién es el responsable de monitorear el Plan.
- Con cuanta frecuencia se realizarán las modificaciones al Plan.
- Como serán evaluados y aprobados los cambios al Plan.
- Como serán realizados y comunicados los cambios al Plan.

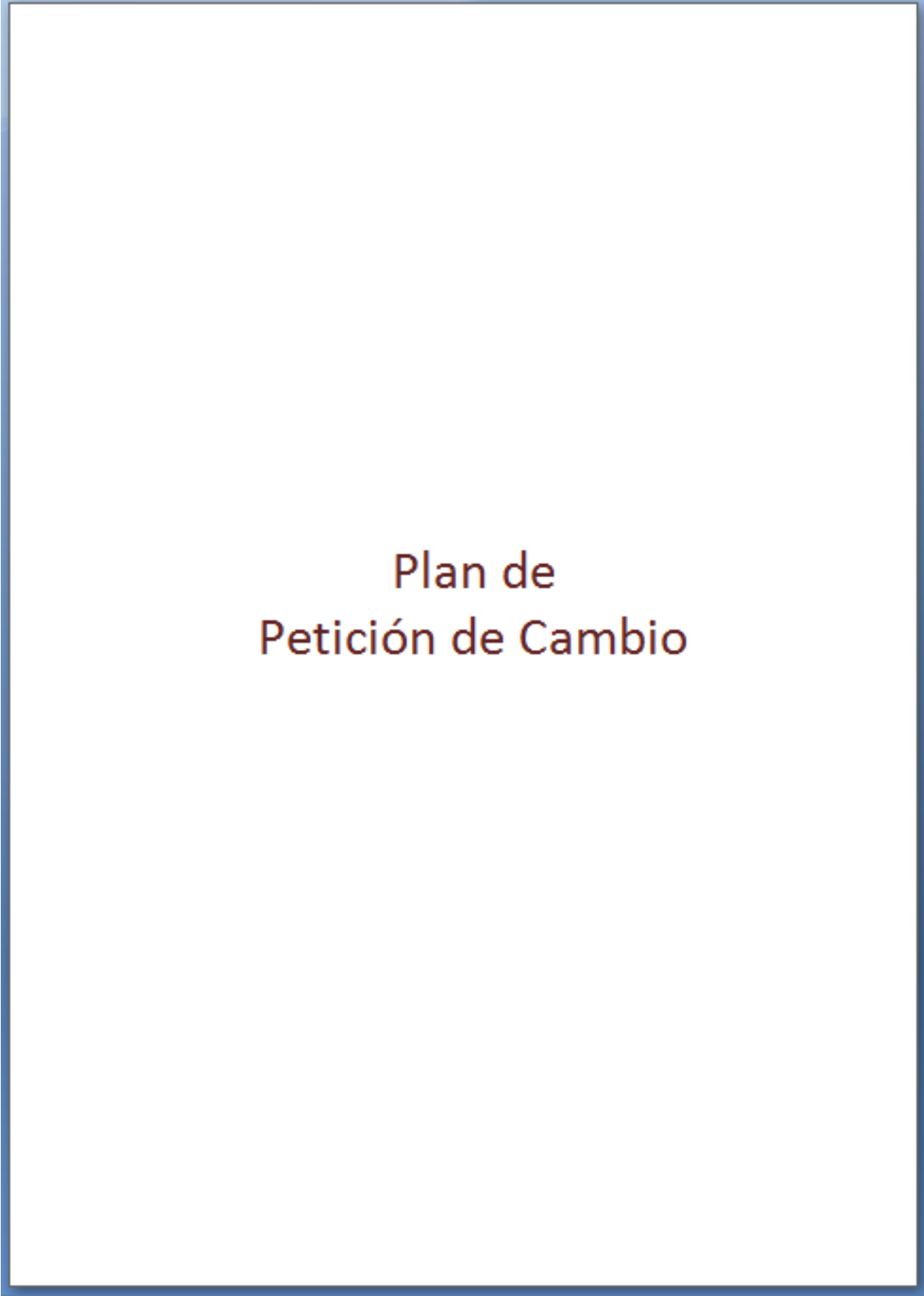
Este plan deberá ser revisado al inicio de cada fase, modificado de acuerdo a lo necesario, aprobado y distribuido al equipo del proyecto.

Figura 69: Producto de Trabajo: Plan de Gestión de Configuración

## ANEXO 7

### Plan de Petición de cambio

## **ANEXO 7: PLAN DE PETICIÓN DE CAMBIO**



# Plan de Petición de Cambio

## [Nombre del proyecto]

### Gestión de Cambios

#### Versión [1.0]

[Este documento es la plantilla base para elaborar el documento Gestión de Cambios. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda.]

#### Historia de revisiones

Fecha	Versión	Descripción	Autor
[dd/mm/aaaa]	[x.x]	[detalles]	[nombre]

#### Tabla de Contenido

Registro de Cambios	2
[Nombre del Producto X]	2
1. Área del producto	2
2. Versión	2
3. Originador	2
4. Fecha	2
5. Urgencia	2
6. Necesidad	2
7. Descripción	2
8. Estado	2
9. Cambios realizados a la Versión Actual	2

### Registro de Cambios

[En esta sección se detallan todos los datos referentes al seguimiento del ciclo de vida de un cambio, que comprende: la solicitud del cambio, la evaluación, la aprobación o rechazo del mismo y la implementación.]

**Nombre del Producto X:**[Nombre del ítem de configuración sobre el cual se solicitó el cambio.]

**1. Área del producto**

[Se establece la Línea de Trabajo o Área a la que corresponde el producto.]

**2. Versión**

[Se indica el cambio en el número de versión del producto.]

Versión Anterior:                      Versión Actual:

**3. Originador**

[Se detalla el nombre y Rol de la persona que da origen al cambio.]

**4. Fecha**

[Se debe especificar la fecha de la solicitud de cambio.]

**5. Urgencia**

[Se detalla la urgencia del cambio en cuestión: ALTA, MEDIA, BAJA]

**6. Necesidad**

[Se detalla la necesidad del cambio solicitado.]

**7. Descripción**

[Se debe describir el cambio que se solicita]

**8. Estado**

[Se actualiza el estado del cambio: SOLICITADO, EVALUADO, APROBADO, RECHAZADO, IMPLEMENTADO]

En caso de ser rechazado se explicarán los motivos a continuación:

Motivo de rechazo:

**9. Cambios realizados a la Versión Actual**

[Se enumeran solamente los objetos de configuración afectados por el cambio realizado, es decir dentro de la versión que se está cambiando cuales son los elementos que cambian de una versión a la siguiente.]

Figura 70: Producto de Trabajo: Plan de Gestión de Cambio

## ANEXO 8

### Plantilla para recoger la información para la auditoria

## ANEXO 8: PLANTILLA PARA RECOGER LA INFORMACIÓN PARA LA AUDITORIA

La auditoria se plantea y se responde con las siguientes preguntas:

AUDITORÍA DE LA CONFIGURACIÓN			
PREGUNTA	DESCRIPCIÓN	SI	NO
¿Se ha hecho el cambio especificado?	Evaluar si el cambio especificado ha sido realizado.		
¿Se han incorporado modificaciones adicionales?	Documentar si la realización del cambio ha supuesto otros cambios no previstos.		
¿Se ha llevado a cabo una revisión técnica formal para evaluar la corrección técnica?	Si se ha realizado una revisión técnica formal para evaluar la corrección del cambio.		
¿Se han seguido adecuadamente los estándares?	Si se ha utilizado para realizar el cambio algún tipo de estándar.		
¿Se han "recalcado" los cambios en el EC?	Si los EC han sufrido los cambios.		
¿Se han especificado la fecha del cambio y el autor?	Registrar la fecha en la que se realiza el cambio además del autor de dicho cambio.		
¿Reflejan los cambios los atributos del objeto de configuración?	Si los cambios realizados reflejan lo esperado.		
¿Se han seguido procedimientos del GC para señalar el cambio, registrarlo y divulgarlo?	Si se ha usado algún procedimiento para señalar, registrar y divulgar el cambio.		
¿Se han actualizado adecuadamente todos los EC relacionados?	Todos los EC relacionados con el cambio deben de haberse actualizado.		

Figura 71: Producto de Trabajo: Plantilla para recoger información de la auditoría



## ANEXO 9

### Plantilla para el informe de auditoria

## ANEXO 9: PLANTILLA PARA EL INFORME DE LA AUDITORIA

INFORME DE AUDITORÍA		
<b>Datos de la Auditoría</b>		
<b>FECHA:</b>	Fecha de realización de la auditoría	
<b>OBJETIVO:</b>	Objetivo de realizar la Auditoría	
<b>PROCESO:</b>	Actividades que se llevan a cabo para realizar la auditoría	
<b>AREA:</b>	Area en la que se realiza la auditoría	
<b>AUDITOR LIDER:</b>	Persona responsable del equipo de la auditoría	
<b>EQUIPO AUDITOR:</b>	Integrantes del equipo de la auditoría	
<b>AUDITADOS:</b>	Quienes son los afectados de la auditoría	
<b>RESUMEN:</b> Descripción del tipo de auditoría que se ha realizado, sobre que se ha realizado, pasos que se han seguido, recursos usados , tiempo empleado y conclusión sobre aquello que está bien y que es lo que se puede mejorar.		
<b>Hallazgos de Auditoría</b>	Que es lo que se ha encontrado en la realización de la auditoría	
<b>Lista de hallazgos</b>	Hacer una lista con lo hallado.	

<b>No Conformidad</b>		
• <b>NO CONFORMIDAD</b>	(no cumplimiento de un requisito del sistema, requiere acción correctiva)	
• <b>DESCRIPCIÓN</b>	Describir la no conformidad encontrada.	
• <b>REQUISITO</b>	Si es necesario algún tipo de requisito para corregir la no conformidad.	
• <b>EVIDENCIA</b>	Describir donde se encuentra la evidencia de la no conformidad.	
<b>Oportunidad</b>		
• <b>OPORTUNIDAD</b>	Cualquier situación que tiene el potencial de convertirse en un problema o un área con potencial de mejora, requiere acción preventiva	
• <b>DESCRIPCIÓN</b>	Describir la situación que puede convertirse en un problema o aquello que necesita una mejora.	

Figura 72: Producto de trabajo: Plantilla para el informe de la auditoría

## ANEXO 10

### Informe para la verificación del código fuente

## **ANEXO 10: INFORME PARA LA VERIFICACIÓN DEL CÓDIGO FUENTE**

### **LISTA DE SUCESOS DE REVISIÓN**

Número revisión: <Corresponde al número de revisión de un producto intermedio>

Fecha: <Fecha en la que se realiza la revisión>

Jefe de revisión: < Persona que realiza la función de jefe en la revisión>

#### **Lista de sucesos**

<Aquí anotaremos todas las pegas que surjan al realizar una revisión así como todos los problemas que se detecten>

<Línea de código1><Suceso 1>

<Línea de código2><Suceso 2>

<Línea de código n><Suceso n>

### **INFORME DE INSPECCIÓN**

#### **Identificación de la revisión**

Proyecto:<Nombre del proyecto>    Número revisión: N

Fecha:<Fecha de revisión>    Lugar: <Nombre del lugar>    Hora: hh:mm:ss

#### **Identificación del producto**

Nombre del producto revisado:<Nombre del producto de trabajo a revisar>

Productor:<Persona que realizó el producto>

Breve descripción:<Descripción del producto de trabajo>

#### **Parte de código a revisar**

Línea de Inicio: <Línea XX>

Línea de Fin: <Línea XX>

<b><u>Material revisado:</u></b>	
<b><u>Equipo de revisión</u></b>	
<b>Nombre</b>	<b>Firma</b>
<Nombre 1>	_____
<Nombre 2>	_____
<Nombre n>	_____
 <b><u>Aprobación del producto</u></b>	
Aceptado: Como esta ( )	
Aceptado con modificaciones menores: <Modificación 1>	
	<Modificación 2>
	<Modificación n>
No aceptado: revisión principal:<Motivos>	
No aceptado: revisión secundaria:<Motivos>	
Revisión no terminada: <Explicación>	
Conclusión:<Explicación de los resultados obtenidos en la revisión>	
 <b><u>Material adicional adjuntado</u></b>	
Lista de sucesos:<Lista realizada anteriormente>	
Materiales usados:<Materiales usados para realizar la revisión>	
Otros (especificar)	

Figura 73: Producto de Trabajo: Informe para verificar el código fuente

## ANEXO 11

### Informe de Inspección

## ANEXO 11: INFORME DE INSPECCIÓN

### INFORME DE INSPECCIÓN

#### Identificación de la revisión

Proyecto: <Nombre del proyecto>

Número revisión: <N>

Fecha: <Fecha de revisión>

Lugar: <Nombre del lugar>

Hora:

Responsable de revisión: <Persona que realiza la función de jefe en la revisión>

#### Identificación del producto

Nombre del producto revisado: <Nombre del producto de trabajo a revisar>

Productor: <Persona que realizó el producto de trabajo que va a ser revisado>

Breve descripción: <Descripción del producto de trabajo que va a ser revisado>

#### Material revisado

##### Tabla de defectos

< Tras realizar el proceso de inspección sobre el producto de trabajo, en este apartado anotaremos todos los defectos que han sido encontradas en el producto de trabajo >

A continuación explicaremos cada uno de los defectos encontrados:

TABLA DE DEFECTOS		
PRODUCTO DE TRABAJO: <Nombre del producto de trabajo>		
Nº DE DEFECTO	EXPLICACIÓN DEL DEFECTO	SOLUCIÓN
1		
2		
3		



### **Equipo de revisión**

En este apartado anotaremos los nombres y apellidos además de la firma de cada integrante del equipo de revisión del producto de trabajo.

EQUIPO DE REVISIÓN		
APELLIDOS	NOMBRE	FIRMA

### **Aprobación del producto**

En este apartado explicaremos en qué estado ha quedado el producto de trabajo tras la inspección. Los diferentes estados en los que se puede encontrar son los siguientes:

- **Aceptado:** El producto de trabajo tras la inspección ha sido aceptado sin ningún defecto encontrado.
- **Aceptado con modificaciones menores:** El producto de trabajo tras la inspección ha sido aceptado pero se han encontrado pequeños defectos que necesitan modificaciones menores para corregirlos. Estas modificaciones serán detalladas en la tabla detallada más abajo y tras realizarlas el producto de trabajo queda aceptado.
- **No aceptado: revisión principal:** El producto de trabajo tras la primera inspección realizada no ha sido aceptado debido a los diferentes motivos. Dichos motivos serán explicados en la tabla detallada más abajo.
- **No aceptado: revisión secundaria:** El producto de trabajo tras realizar la segunda inspección sobre él, no ha sido aceptado debido a los diferentes motivos que serán explicados en la tabla detallada más abajo.
- **Revisión no terminada:** Explicación de los motivos por los cuales la inspección sobre el producto de trabajo no ha sido finalizada.
- **Conclusión:** Explicación de los resultados obtenidos en la inspección del producto de trabajo.

APROBACIÓN DEL PRODUCTO DE TRABAJO				
ACEPTADO SIN MODIFICACIONES	ACEPTADO CON MODIFICACIONES	NOACEPTADO (REVISIÓN PPAL)	NO ACEPTADO (REVISIÓN SECUNDARIA)	REVISIÓN NO FINALIZADA
A continuación, según la opción marcada anteriormente, rellenaremos el correspondiente apartado:				
ACEPTADO CON MODIFICACIONES	Descripción que refleja en que ha consistido cada una de las modificaciones que han sido realizadas sobre nuestro producto de trabajo: <ul style="list-style-type: none"> <li>• Modificación 1: &lt;descripción&gt;</li> <li>• Modificación 2: &lt;descripción&gt;</li> <li>• Modificación n: &lt;descripción&gt;</li> </ul>			
NO ACEPTADO (REVISIÓN PPAL)	Descripción de cada uno de los motivos por los cuales nuestro producto de trabajo no ha sido aceptado en la revisión principal: <ul style="list-style-type: none"> <li>• Motivo 1: &lt;descripción&gt;</li> <li>• Motivo 2: &lt;descripción&gt;</li> <li>• Motivo n: &lt;descripción&gt;</li> </ul>			
NO ACEPTADO (REVISIÓN SECUNDARIA)	Descripción de cada uno de los motivos por los cuales nuestro producto de trabajo no ha sido aceptado en la revisión secundaria: <ul style="list-style-type: none"> <li>• Motivo 1: &lt;descripción&gt;</li> <li>• Motivo 2: &lt;descripción&gt;</li> <li>• Motivo n: &lt;descripción&gt;</li> </ul>			
REVISIÓN NO FINALIZADA	Explicación de los motivos por los cuales la revisión de nuestro producto de trabajo no ha sido finalizada.			
CONCLUSIÓN	Evaluación de los resultados obtenidos en la inspección del producto de trabajo, en función del número de defectos que han sido encontrados, si se trata de defectos graves o fáciles de subsanar.			

#### Material adicional adjuntado

En este apartado anotaremos todo aquel material que ha sido usado para poder realizar la inspección del producto de trabajo.

**Materiales usados:** <materiales usados para realizar la inspección>

Figura 74: Producto de Trabajo: Informe de inspección

## ANEXO 12

### Informe de Verificación

## ANEXO 12: INFORME DE VERIFICACIÓN

### INFORME DE VERIFICACIÓN

#### Identificación de la revisión

En este apartado rellenaremos los diferentes campos relacionados con la revisión.

Proyecto:<Nombre del proyecto>      Número verificación: <corresponde al número de verificación de un procedimiento>

Fecha:<fecha de revisión>      Lugar:      Hora:

Responsable de revisión: < Persona que realiza la función de jefe en la verificación>

#### Identificación del producto de trabajo

En este apartado rellenaremos los diferentes campos relacionados con el producto de trabajo seleccionado para verificar sus actividades.

Nombre del producto de trabajo :<nombre del producto de trabajo del cual se van a verificar las actividades>

Productor:<persona que realizó el producto de trabajo>

Breve descripción:<descripción del producto de trabajo>

#### Identificación de las actividades del producto de trabajo

En este apartado identificaremos todas las actividades que componen el producto de trabajo. Para ello se rellenará la siguiente tabla en la que aparece el número de la actividad, su nombre y una breve descripción que explica en que consiste la actividad.

ACTIVIDADES DEL PRODUCTO DE TRABAJO		
PRODUCTO DE TRABAJO: <Nombre del producto de trabajo>		
ACTIVIDAD	NOMBRE DE LA ACTIVIDAD	DESCRIPCIÓN
Nº 1		
Nº 2		

**Material revisado:**

**Tabla de defectos**

Tras realizar el proceso de verificación sobre las actividades del producto de trabajo, en este apartado anotaremos todos los defectos que han sido encontradas.

A continuación explicaremos cada uno de los defectos encontrados:

TABLA DE DEFECTOS		
PRODUCTO DE TRABAJO:<Nombre del producto de trabajo>		
EN QUÉ ACTIVIDAD	EXPLICACIÓN DEL DEFECTO	SOLUCIÓN
<nombre de la actividad>	<descripción del defecto encontrado>	<explicación para solucionar el defecto>

**Equipo de revisión**

En Este apartado anotaremos los nombres y apellidos además de la firma de cada integrante del equipo de revisión del producto de trabajo.

EQUIPO DE REVISIÓN		
APELLIDOS	NOMBRE	FIRMA

### **Aprobación de las actividades del producto de trabajo**

En este apartado explicaremos en qué estado ha quedado el producto de trabajo tras la inspección. Los diferentes estados en los que se puede encontrar son los siguientes:

**Aceptado:** Las actividades del producto de trabajo tras la verificación han sido aceptadas sin ningún defecto encontrado.

**Aceptado con modificaciones menores:** Las actividades del producto de trabajo tras la verificación han sido aceptadas pero se han encontrado pequeños defectos que necesitan modificaciones menores para corregirlos. Estas modificaciones serán detalladas en la tabla detallada más abajo y tras realizarlas dichas actividades son aceptadas.

**No aceptado: revisión principal:** Las actividades del producto de trabajo tras la primera verificación realizada no ha sido aceptado debido a los diferentes motivos. Dichos motivos serán explicados en la tabla detallada más abajo.

**No aceptado: revisión secundaria:** Las actividades del producto de trabajo tras realizar la segunda inspección sobre él, no ha sido aceptado debido a los diferentes motivos que serán explicados en la tabla detallada más abajo.

**Revisión no terminada:** Explicación de los motivos por los cuales la verificación sobre las actividades del producto de trabajo no ha sido finalizada.

**Fortalezas y debilidades encontradas:** Explicación de las fortalezas y debilidades que se han encontrado en la verificación de las actividades del producto de trabajo

APROBACIÓN DE LAS ACTIVIDADES DEL PRODUCTO DE TRABAJO				
ACEPTADO SIN MODIFICACIONES	ACEPTADO CON MODIFICACIONES	NO ACEPTADO (REVISIÓN PPAL)	NO ACEPTADO (REVISIÓN SECUNDARIA)	VERIFICACION NO FINALIZADA
A continuación, según la opción marcada anteriormente, rellenaremos el correspondiente apartado:				
ACEPTADO CON MODIFICACIONES	Descripción de en qué ha consistido cada una de las modificaciones que han sido realizadas sobre las actividades del producto de trabajo: <ul style="list-style-type: none"> <li>• Modificación 1: &lt;descripción de la modificación e indicar en qué actividad ha sido realizada&gt;</li> <li>• Modificación 2: &lt;descripción&gt;</li> <li>• Modificación n: &lt;descripción&gt;</li> </ul>			
NO ACEPTADO (REVISIÓN PPAL)	Descripción de cada uno de los motivos por los cuales la actividad no ha sido aceptada en la principal: <ul style="list-style-type: none"> <li>• Motivo 1: &lt;descripción&gt;</li> <li>• Motivo 2: &lt;descripción&gt;</li> <li>• Motivo n: &lt;descripción&gt;</li> </ul>			
NO ACEPTADO (REVISIÓN SECUNDARIA)	Descripción de cada uno de los motivos por los cuales la actividad no ha sido aceptada en la verificación secundaria: <ul style="list-style-type: none"> <li>• Motivo 1: &lt;descripción&gt;</li> <li>• Motivo 2: &lt;descripción&gt;</li> <li>• Motivo n: &lt;descripción&gt;</li> </ul>			
VERIFICACION NO FINALIZADA	Explicación de los motivos por los cuales la verificación de las actividades del producto de trabajo no ha sido finalizada.			
FORTALEZAS Y DEBILIDADES	Explicación de las fortalezas y debilidades que se han encontrado en la verificación de las actividades del producto de trabajo			

#### Material adicional adjuntado

En este apartado anotaremos todo aquel material que ha sido usado para poder realizar la verificación de las actividades del producto de trabajo.

**Materiales usados:** <materiales usados para realizar la verificación>

Figura 75: Producto de Trabajo: Informe de verificación

## ANEXO 13

### Informe de Resultados



## ANEXO 13: INFORME DE RESULTADOS

### INFORME DE RESULTADOS

#### Identificación del informe

Proyecto:<Nombre del proyecto> Número de informe: <Número de informe>

Fecha:<Fecha del informe> Lugar: <Nombre del lugar> Hora: hh:mm:ss

#### Identificación del producto de trabajo

Nombre del producto de trabajo :<nombre del producto de trabajo que se ha inspeccionado>

Responsable:<persona que realizó el producto de trabajo>

#### Identificación de las actividades del producto de trabajo

En este apartado identificaremos todas las actividades que componen el producto de trabajo. Para ello se rellenará la siguiente tabla en la que aparece el número de la actividad, su nombre y una breve descripción que explica en qué consiste la actividad.

ACTIVIDADES DEL PRODUCTO DE TRABAJO		
PRODUCTO DE TRABAJO: <Nombre del producto de trabajo>		
ACTIVIDAD	NOMBRE DE LA ACTIVIDAD	DESCRIPCIÓN
Nº 1		
Nº 2		

#### Problemas y causas encontrados en el producto de trabajo

TABLA DE DEFECTOS			
PRODUCTO DE TRABAJO:<Nombre del producto de trabajo>			
Nº DE DEFECTO	EXPLICACIÓN DEL DEFECTO	CAUSA DEL DEFECTO	SOLUCIÓN
Nº 1			
Nº 2			
Nº 3			

**Problemas y causas encontrados en las actividades del producto de trabajo**

TABLA DE DEFECTOS			
PRODUCTO DE TRABAJO:<Nombre del producto de trabajo>			
EN QUÉ ACTIVIDAD <nombre de la actividad>	EXPLICACIÓN DEL DEFECTO <descripción del defecto encontrado>	CAUSA	SOLUCIÓN <explicación para solucionar el defecto>

**Lecciones aprendidas**

<Explicación de que lecciones se ha aprendido de todos los errores cometidos durante el proceso de verificación>

**Prácticas excelentes:** < Determinar todo lo que se ha realizado correctamente>

**Prácticas deficientes:**< Determinar todo lo que se ha realizado mal>

**Riesgos altos:** < Determinar si se ha producido algún tipo de riesgo>

Figura 76: Producto de Trabajo: Informe de resultados